(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平11－249995

(43)公開日　平成11年(1999)９月17日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　13/00	３５４	Ｇ０６Ｆ　13/00	３５４Ｄ
17/30		15/40	３１０Ｆ
			３３０Ｚ

審査請求　未請求　請求項の数１　ＯＬ　外国語出願　（全111頁）

(21)出願番号　特願平10－316771

(22)出願日　平成10年(1998)10月５日

(31)優先権主張番号　０８／９４４，１２１
(32)優先日　1997年10月６日
(33)優先権主張国　米国（ＵＳ）
(31)優先権主張番号　０８／９４４，１２４
(32)優先日　1997年10月６日
(33)優先権主張国　米国（ＵＳ）
(31)優先権主張番号　０８／９４４，１２５
(32)優先日　1997年10月６日
(33)優先権主張国　米国（ＵＳ）

(71)出願人　592089054
エヌシーアール　インターナショナル　インコーポレイテッド
ＮＣＲ　Ｉｎｔｅｒｎａｔｉｏｎａｌ，Ｉｎｃ．
アメリカ合衆国　45479　オハイオ、デイトン　サウス　パターソン　ブールバード　1700

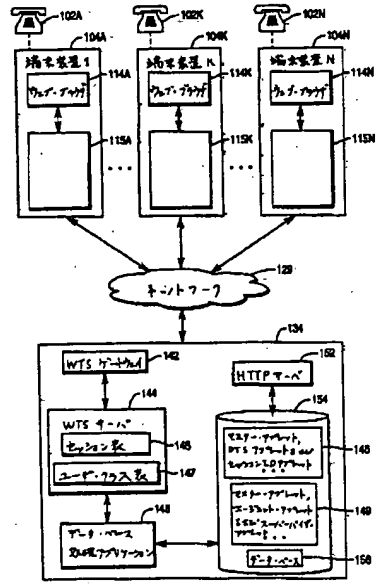
(72)発明者　ジェイムス　エイ　シェルトン
アメリカ合衆国　ニュージャージー州　07733ヘイワードヒルス　ドライブ　ホルムデル　21

(74)代理人　弁理士　西山　善章

最終頁に続く

(54)【発明の名称】　インターネットのウエブサイトへのアクセスをコーディネイトするための方法および装置

(57)【要約】　（修正有）
【課題】　端末装置のアクティビティを管理するための方法を提供する。
【解決手段】　端末装置は、ＨＴＴＰサーバ１５２からページを検索するが、各ページはアプレットのような埋め込まれた情報を持つ。ある端末装置において行われたロード、アンロードまたはデータの変更等のページ・アクティビティに応じて、各アプレットは、サーバ１４４にページＵＲＬと一緒に前記アクティビティを報告し、サーバはアクティビティをデータベース１４８に記憶する。異なる端末装置におけるページ間の同期は、データ追跡および同期アプレットのような適当な埋め込み情報により行われる。アプレットは、アクティビティを識別し、必要に応じて、データベースに記憶し、また他の参加端末装置へ送るために、アクティビティの詳細を端末装置を通して追跡サーバに転送する。



【特許請求の範囲】
【請求項1】　端末装置［104A］において、ページ［204、214、224、．．．］に対して行われたアクティビティと、前記端末装置と他の端末装置［104K、104N］との間のアクティビティを管理する方法であって、前記ページがネットワーク・サイト［134］から検索中のものであり、
（a）前記端末装置において、ページを検索するステップと、
（b）前記端末装置において、検索された前記ページに対するアクティビティ［114A、115A］を行うステップと、
（c）ネットワーク・サイトにおいて、前記端末装置に対する前記アクティビティを記録［148］するステップと、
（d）前記他の端末装置において、前記端末装置との通信を確立するステップと、（e）前記他の端末装置において、前記端末装置に対して記録したアクティビティを表示するステップと、を含むことを特徴とするインターネットのウエブサイトへのアクセスをコーディネイトするための方法。
【発明の詳細な説明】
【0001】
【発明の属する技術分野】本発明は、概して、ユーザ端末装置の一つのグループで実行されている、一つのグループのウェブ・ブラウザによるインターネット・ウェブ・サイトへのアクセスを調整するための方法および装置に関する。
【0002】
【従来の技術】ユーザがウェブ・サイト（ネットワーク・サイト）からインターネットを通して情報を検索することができることは周知である。ウェブ・サイトから情報を検索するための基本的なモデルは、ユーザ始動の情報捜索である。より詳細に説明すると、ユーザは、ウェブ・サイトに要求を送るために、（端末装置を通して）ウェブ・ブラウザと交信する。前記要求に応えて、ウェブ・ブラウザに対してウェブ・サーバは、要求された情報を検索し、ウェブ・ブラウザに、いわゆるウェブ・ページ（HTML）フォーマットに従って配列された情報を送る。このモデルの独特の特徴の一つは、検索されたウェブ・ページ内に埋め込まれた「ハイパーテキスト・リンク」機能である。この機能により、ユーザは情報を検索する際に、あるウェブ・ページから他のウェブ・ページに「ナビゲート」することができる。インターネットを通してユーザ（または顧客）にサービス（または助力）を提供するために、あるグループのブラウザの間でウェブ・ページに行われたアクティビティを追跡する機構を提供することが望ましい。
【0003】
【発明が解決しようとする課題】ナビゲートされたウェブ・ページを追跡する一つの方法は、ウェブ・サイトに監視プログラムを設置することである。ある端末装置がウェブ・サイトに要求を送ると、ウェブ・サイトの監視プログラムが、要求されたウェブ・ページに対するURL（Uniform Resource Locator、以下「URL」という）を収集し、前記URLをサーバに送る。しかし、この方法の場合には、監視プログラムは、端末装置からの要求をいつでも監視することができない。何故なら、端末装置がそのブラウザのキャッシュ・スペースまたはプロキシ・サーバからのウェブ・ページを検索しているときには、前記要求が局地的に実行され、ウェブ・サイトに決して送られないからである。その結果、URLを正確に追跡することができない。
【0004】ナビゲートされたウェブ・ページを追跡するための他の方法は、端末装置のところに、ブラウザと一緒に監視プログラムを設置することである。前記監視プログラムは、絶えずウェブ・ブラウザと通信する。しかし、あるベンダーが製造したウェブ・ブラウザ用に設計された監視プログラムは、通常、他のベンダーが製造したウェブ・ブラウザと相互に動作することもできなければ、前記ウェブ・ブラウザのところへ持って行くこともできない。何故なら、ブラウザ・インターフェース機構は専有タイプであるからである。その結果、ユーザは複雑な監視プログラムを使用しなければならないことになる。
【0005】それ故、ウェブ・ページを追跡し、ウェブ・ナビゲーション・ソフトウェアについての深い知識を必要としない技術が必要になる。前記技術は、それ自身のウェブ・ページ・アクティビティが、ウェブ・サイトにそのアクティビティを報告する要求を発生しない場合でも、動作する必要がある。
【0006】本発明は、ある端末装置のところで、ネットワーク・サイトから検索されるページに対して行われたアクティビティ、および前記端末装置と他の端末装置との間のアクティビティを管理するための方法である。前記方法は、（a）前記端末装置のところで、ページを検索するステップと、（b）前記端末装置のところで、前記検索したページに対してアクティビティを行うステップと、（c）ネットワーク・サイトのところで、前記端末装置に対する前記アクティビティを記録するステップと、（d）前記他の端末装置のところで、前記端末装置との連絡を確立するステップと、（e）他の端末装置のところで、前記端末装置に対するアクティビティ記録を表示するステップとからなる。
【0007】本発明は、また前記端末装置または前記他の端末装置のところで行われたアクティビティの間でページの同期を管理する方法でもある。前記アクティビティは、ページのロード、ページのアンロード、およびページ上のデータ・フィールドの変更を含むことができる。

【0008】他の端末装置としては、管理端末装置を使用することができる。

【0009】本発明は、ページ追跡サーバへのアクティビティについての情報を送ることによって、個々の端末装置に対する、ネットワーク・サイトにおけるページ・サーバからの検索によりネットワークを通してロードすることを要求されたページに対するアクティビティを追跡するための方法でもある。前記方法は、（ｇ）前記サーバ内のページの位置を示すためのページ・ロケータに関連し、あるプログラムを表示するための少なくとも位置情報のプログラム情報を含む第一のページをページ・サーバからロードするステップと、（ｈ）前記位置情報に基づいて、サーバから前記プログラムをロードし、そのプログラムを実行するステップと、（ｊ）前記ページに関するアクティビティをプログラムにより監視するステップと、（ｋ）その内部に記憶できるように、ページ追跡サーバに、監視したアクティビティに関する情報をプログラムにより送るステップとを含む。前記プログラム情報は、実際のプログラムである場合もある。前記プログラム・ロケータは、ＵＲＬである場合もあるし、前記プログラム情報はタグである場合もある。

【0010】

【発明の実施の形態】下記の説明は、当業者が本発明を使用することができるようにするためのものであり、特定の用途および要件に関連するものである。当業者であれば、上記好適な実施形態の種々の修正をすぐに思い浮かべることができるだろうし、本特許明細書に記載した原理は、本発明の精神および範囲から逸脱することなしに他の実施形態および用途に適用することができることを理解することができるだろう。それ故、本発明は、上記実施形態により制限されるものではなく、本特許明細書に開示する原理および特徴と一致する広い範囲で解釈すべきものである。

【0011】図１について説明すると、この図は、本発明の例示としてのウェブ・ページ同期システム１００である。

【0012】図１に示すように、前記システムは、Ｎ台の端末装置（１０４Ａ、１０４Ｋ、および１０４Ｎ）、ネットワーク１２９（インターネット、または前記インターネットと、あるインターネットとの組合せ）およびウェブ・サイト１３４を含む。各端末装置は、その付近に設置された電話セット（１０２Ａ、１０２Ｋ、または１０２Ｎ）を持つ。各端末装置は、パソコンでも、ワーク・ステーションでも、ジャバ・ステーションでも、またはウェブＴＶシステムであってもよい。

【0013】ウェブ・サイト１３４は、ＷＴＳ（ウェブ追跡および同期）ゲートウェイ１４２、セッション表１４５およびユーザ・クラス表１４７を含むＷＴＳサーバ１４４、データベース処理アプリケーション１４８、ＨＴＴＰ（Hyper Text Transfer Protocol、以下「ＨＴＴＰ」という）サーバ１５２、消費者ページ貯蔵所を記憶するためのハード・ディスク・ユニット１５４、消費者ページ貯蔵所１４６、管理ページ貯蔵所１４９、およびデータベース１５６を含む。ウェブ・サイト１３４内のすべての構成部分は、一つまたはそれ以上のコンピュータ・システムに設置することができる。各コンピュータ・システムは、処理ユニット（複数のプロセッサを含むことができる）、メモリ装置、およびディスク・ユニット（複数のディスク・セットを含むことができる）を含む。

【0014】各端末装置（１０４Ａ、１０４Ｋ、または１０４Ｎ）は、プロセッサ・ユニット（図示せず）およびメモリ領域（１１５Ａ、１１５Ｋ、および１１５Ｎ）を含み、ジャバ・イネーブル・ウェブ・ブラウザ（１１４Ａ、１１４Ｋ、または１１４Ｎ）を実行する。各メモリ領域（１１５Ａ、１１５Ｋ、および１１５Ｎ）は、その各ブラウザ（１１４Ａ、１１４Ｋ、または１１４Ｎ）により維持される。ネットワーク１２９を通して、各ブラウザ（１１４Ａ、１１４Ｋ、または１１４Ｎ）は、ＨＴＴＰサーバ１５２に要求を送ることができ、また前記サーバからウェブ・ページを受信することができ、受信したウェブ・ページをその各端末装置のところで表示することができる。各ブラウザ（１１４Ａ、１１４Ｋ、または１１４Ｎ）は、マスター・アプレット（１２４Ａ、１２４Ｋ、または１２４Ｎ）、一組のＤＴＳ（データ追跡および同期）アプレット、セッションＩＤアプレット、およびエージェント・アプレットを実行することができる。図１に示すように、前記アプレットは、消費者ページ貯蔵所１４６に記憶され、前記消費者ページ貯蔵所１４６からダウンロードし、端末装置（１０４Ａ、１０４Ｋ、または１０４Ｎ）のメモリ領域に記憶することができる。

【0015】図２について説明すると、この図は、本発明により、各端末装置（１０４Ａ、１０４Ｋ、または１０４Ｎ）が、そのマスター・アプレット（１２４Ａ、１２４Ｋ、または１２４Ｎ）ＤＴＳアプレット（１２６Ａ、１２６Ｋまたは１２８Ｎ）およびセッションＩＤアプレット（１２８Ａ、１２８Ｋ、１２８Ｎ）をダウンロードしたときの状態を示す。

【0016】図２においては、各（消費者）マスター・アプレット（１２４Ａ、１２４Ｋ、または１２４Ｎ）は、主として、（１）その各ブラウザのところで各ウェブ・ページがロードされたとき、それに応じて、専用ソケットを開き、その各ブラウザ（１１４Ａ、１１４Ｋ、または１１４Ｎ）に対して、ネットワーク１２９を通してＷＴＳゲートウェイ１４２へのソケットの接続を確立することと、（２）そこから送られてくるコマンドおよび情報のソース（例えば、どのブラウザ、どのウェブ・ページ等）を識別することができるＷＴＳサーバ１４４、ソケット接続を通してＷＴＳサーバ１４４と通信す

ることと、（３）その各ブラウザのアクティビティを監視することと、（４）その各ブラウザのアクティビティについての情報をＷＴＳサーバ１４４に送ることと、（５）他のブラウザのアクティビティについての情報を受信し、処理することと、（６）ソケット接続を通して、ＤＴＳアプレット（１２６Ａ、１２６Ｋ、または１２６Ｎ）、セッションＩＤアプレット（１２８Ａ、１２８Ｋ、または１２８Ｎ）、またはマスター・アプレットと一緒に同じページに埋め込まれている任意の他の消費者アプレット用のＷＴＳサーバ１４４への一本の通信経路を供給することと、（７）それ自身およびＤＴＳアプレット（１２６Ａ、１２６Ｋ、または１２６Ｎ）、セッションＩＤアプレット（１２８Ａ、１２８Ｋ、または１２８Ｎ）、またはマスター・アプレットと一緒に、同じページに埋め込まれている、任意の他の消費者アプレットに対すサービスを要求するために、ＷＴＳサーバ１４４にコマンドを送ることと、（８）その各ブラウザが消費者ユーザであることを表示するために、コマンドと一緒にユーザ・クラス情報を送ることとに主として責任を持つ。

【００１７】ＤＴＳアプレット（１２６Ａ、１２６Ｋ、または１２６Ｎ）の各組は、（１）その各ブラウザにより現在表示されているウェブ・ページ上でのデータ・アクティビティ（データ入力またはデータ・フィールドのデータ更新）の表示と監視、（２）その各マスター・アプレットを通してのＷＴＳサーバ１４４へのデータ・アクティビティの送信、（３）その各マスター・アプレットを通しての他のブラウザからのデータ・アクティビティの受信、（４）その各ブラウザにより現在表示されているウェブ・ページに対する他のブラウザからのデータ・アクティビティの処理に対して主として責任を持つ、一つまたはそれ以上の個々のＤＴＳアプレットを含む。

【００１８】各セッションＩＤアプレット（１２８Ａ、１２８Ｋ、または１２８Ｎ）は、ウェブ・ページ上での現在のセッションＩＤの検索および表示に対して責任を持つ。

【００１９】管理ページ貯蔵所１４９に示すように、エージェント・アプレット（またはスーパーバイザ・アプレット）は、セッション・インターフェースの生成、前記セッション・インターフェースを通してのセッションへの参加、監視および制御に対して責任を持つ。（管理）マスター・アプレットは、（１）専用ソケットを開くこと、エージェント・アプレット、スーパーバイザ・アプレット、またはマスター・アプレットと一緒に、同じウェブ・ページに埋め込まれている任意の他の管理アプレットが生成したセッション・インターフェースに対して、ネットワーク１２９を通してＷＴＳゲートウェイ１４２へのソケット接続の確立、（２）そこから、ＷＴＳサーバ１４４が、送られてくるコマンドおよび情報のソース（例えば、どのブラウザ、どのウェブ・ページ等）を識別することができる、ソケット接続を通してＷＴＳサーバ１４４と通信することと、（３）ソケット接続を通して、エージェント・アプレットまたはマスター・アプレットと一緒に、同じページに埋め込まれている任意の他の管理アプレットに対して、ＷＴＳサーバ１４４への一本の通信経路を供給することと、（４）その各ブラウザが、管理ユーザであることを表示するために、コマンドと一緒にユーザ・クラス情報を送ることに主として責任を持つ。

【００２０】ＷＴＳゲートウェイ１４２は、マスター・アプレットとＷＴＳサーバ１４４との間のすべてのソケット接続の維持に責任を持つ。マスター・アプレットとＷＴＳゲートウェイ１４２の間の接続は、標準ソケットを使用することによって行われる。ＷＴＳゲートウェイ１４２とＷＴＳサーバ１４４との間の接続は、ＲＭＩ（遠隔方法革新）を使用して行われる。

【００２１】ＷＴＳサーバ１４４は、（１）能動セッション、ウェブ・ページのロード、相互作用およびアンロードを含む、例示としてのアクティビティに参加するすべてのブラウザのアクティビティの管理と追跡、（２）前記アクティビティについての情報の記録、（３）前記能動セッションに参加するすべてのブラウザに対するアクティビティの同期の管理、（４）（ブラウザを通して）消費者ユーザが、最初にウェブ・サイト１３４に要求を送ったときのセッションの生成、（５）セッションの長さの定義、（６）特定のセッションの長さより長い時間能動状態にないセッションの除去、（７）あるセッションへの参加者の追加と削除、（８）（消費者）マスター・アプレット、ＤＴＳアプレット、セッションＩＤアプレットのような消費者アプレット、および（管理）マスター・アプレット、エージェント・アプレットおよびスーパーバイザ・アプレットのような管理アプレットからのすべてのコマンドに対するサービスの提供に対して責任を持つ。

【００２２】消費者ページ貯蔵所１４６は、消費者に対するウェブ・ページおよびアプレットを記憶する。消費者アプレットは、消費者ウェブ・ページに選択的に埋め込むことができる。例示としての消費者アプレットは、（消費者）マスター・アプレット、ＤＴＳアプレット、セッションＩＤアプレット等を含む。

【００２３】管理ページ貯蔵所１４９は、アドミニストレータ、スーパーバイザ、エージェント等を含む、呼出センター管理ユーザ用のウェブ・ページおよびアプレットを記憶する。管理アプレットは、管理ウェブ・ページに選択的に埋め込むことができる。例示としての管理アプレットは、（管理）マスター・アプレット、エージェント・アプレット、スーパーバイザ・アプレット等を含む。

【００２４】本発明をもっとはっきりと説明するために、貯蔵所１４６に記憶している（または、からダウン

ロードした）アプレットは、消費者アプレットと呼ぶことができ、貯蔵所１４９に記憶している（または、貯蔵所１４９からダウンロードした）アプレットは、管理アプレットと呼ぶことができる。例えば、貯蔵所１４６に記憶されている（または、貯蔵所１４６からダウンロードされた）マスター・アプレットは、消費者マスター・アプレットと呼ぶことができ、貯蔵所１４９に記憶されている（または、貯蔵所１４９からダウンロードされた）マスター・アプレットは、管理マスター・アプレットと呼ぶことができる。ＨＴＴＰサーバ１５２は、消費者ユーザが、消費者ページ貯蔵所１４６に記憶されている、ウェブ・ページにだけアクセスすることができ、また管理ユーザ（アドミニストレータ、スーパーバイザ、エージェント等）が、消費者ページ貯蔵所１４６および管理ページ貯蔵所１４９の両方に記憶しているウェブ・ページにアクセスすることができる機密保護アプリケーションを含む。

【００２５】セッション表１４５は、すべての能動セッションに対する情報を維持する責任を持つ。

【００２６】クラス表１４７は、異なるユーザに割り当てられたユーザ・クラスの記録を保管する責任を持つ。例示としてのユーザ・クラスとしては、アドミニストレータ、スーパーバイザ、エージェントおよび消費者がある。

【００２７】ユーザ・クラス（アドミニストレータ、スーパーバイザ、エージェントおよび消費者）に基づいて、ＷＴＳサーバ１４４は、下記のサービスを提供する。

（１）セッション（消費者）の生成

（２）セッション参加者（スーパーバイザ、エージェントおよび消費者）から受信したデータの記憶

（３）能動セッション（アドミニストレータおよびスーパーバイザ）のリストの作成

（４）能動セッション（アドミニストレータおよびスーパーバイザ）に関連する情報のリストの作成

（５）現在のユーザ（アドミニストレータ）のリストの作成

（６）セッション（スーパーバイザおよびエージェント）の参加

（７）セッション（スーパーバイザ）の終了

（８）セッション（スーパーバイザおよびエージェント）の監視

（９）セッション・パラメータ（アドミニストレータ）の構成

（１０）参加ブラウザ（スーパーバイザ、エージェントおよび消費者）の消費者マスター・アプレット、または管理マスター・アプレットへのコマンドおよび情報の送信

データベース１５６は、セッション表１４５内に収集したデータの記憶に対して責任を持つ。

【００２８】ＨＴＴＰサーバ１５２は、ウェブ・ブラウザの一人が発行した要求の処理、消費者ページ貯蔵所１４６または管理ページ貯蔵所１４９からのウェブ・ページの検索、および前記要求を発生したブラウザへのウェブ・ページの送信に対して責任を持つ。

【００２９】データベース処理アプリケーション１４８は、セッション表１４５内の収集したデータのデータベース１５６への書き込みに対して責任を持つ。

【００３０】図３について説明すると、この図は、最初のウェブ・ページのロードに応じて、ＨＴＴＰサーバから端末装置１０４Ａへの（消費者）マスター・アプレット、ＤＴＳアプレットおよびセッションＩＤアプレット１５２のダウンロードの仕方、およびその後の本発明の動作を行うための呼出し方法のプロセスである。

【００３１】図３に示すように、（消費者）マスター・アプレット、一組のＤＴＳアプレット、およびセッションＩＤアプレットは、一組のアプレット・タグ２０８を使用することによりウェブ・ページ２０４に埋め込まれる。ウェブ・ページ２０４は、ＨＴＴＰサーバ１５２内のウェブ・ページ２０４の位置を示す特定のＵＲＬに関連する。

【００３２】点線（１）で示すように、ウェブ・ブラウザ１１４Ａは、ウェブ・ページ２０４のＵＲＬを含む要求を、ネットワーク１２９を通してＨＴＴＰサーバ１５２に送る。点線（２）で示すように、前記要求に応えて、ＨＴＴＰサーバ１５２は、消費者ページ貯蔵所１４６からのウェブ・ページを検索し、それをネットワーク１２９を通してウェブ・ブラウザ１１４Ａに送る。ウェブ・ページ２０４は、ＨＴＴＰサーバ１５２内のマスター・アプレット、ＤＴＳアプレット、およびセッションＩＤアプレットの位置を示す一組のアプレット・タグ２０８を含む。点線（３）で示すように、ウェブ・ブラウザ１１４Ａは、ウェブ・ページ２０４をロードする。点線（４）で示すように、マスター・アプレット、ＤＴＳアプレットおよびセッションＩＤアプレットは、まだダウンロードされていないので、ウェブ・ブラウザ１１４Ａは、ネットワーク１２９を通して、アプレット・タグ２０８に基づいて、前記アプレットをダウンロードするよう要求を送る。点線（５）で示すように、ＨＴＴＰサーバ１５２は、マスター・アプレット、ＤＴＳアプレット、およびセッションＩＤアプレットを、ネットワーク１２９を通してブラウザ１１４Ａに送る。点線（６）で示すように、ブラウザ１１４Ａはマスター・アプレット１２４Ａ、ＤＴＳアプレット１２６Ａ、およびセッションＩＤアプレット１２８Ａを端末装置１０４Ａのメモリ領域１１５Ａに記憶し、初期化し、前記アプレットを呼び出す。呼び出した後で、前記アプレットが責任を持つように割り当てられているアクティビティを監視し、処理するために、前記アプレットは、ウェブ・ブラウザ１１４Ａと一緒に動作する。点線（７）で示すように、マ

スター・アプレット１２４Ａは、専用ソケットを開き、ブラウザ１１４Ａおよびウェブ・ページ２０４に対するＷＴＳゲートウェイ１４２へのソケットの接続を確立する。ソケット接続を通して、マスター・アプレット１２６は、ブラウザ１１４Ａに対する一意のＩＤと一緒に、コマンドをＷＴＳサーバ１４４に送る。マスター・アプレット１２６からのコマンドに応じて、ＷＴＳサーバ１４４は、前記一意のＩＤに基づいて、ブラウザ１１４０Ａに対するセッションを生成し、前記コマンドを受信した時間を示すタイム・スタンプ（ロード時間）を発行し、ウェブ・ページ２０４のＵＲＬおよびタイム・スタンプをブラウザ１１４のために生成されたセッションに記憶する。図６を参照しながら、以下の説明を読めば分かるように、ＵＲＬ、コマンドおよびロード時間は、前記セッションに対して生成されたＵＲＬ経歴リストおよびコマンド・リストに記憶される。

【００３３】図４について説明すると、この図は、マスター・アプレット１２４Ａ、ＤＴＳアプレット１２６Ａ、およびセッションＩＤアプレット１２８Ａが、端末装置１０４Ａにすでにダウンロードされ、キャッシュに記憶されている場合には、本発明の動作を行うために、（ウェブ・ページ２０４から後ろの）以降のウェブ・ページ２１４のロードに応じて、前記セッションＩＤアプレット１２８Ａが呼び出されるプロセスである。

【００３４】点線（１）で示すように、ウェブ・ページ２１４をダウンロードするために、ウェブ・ブラウザ１１４Ａは、ウェブ・ページ２１４のＵＲＬを含む要求を、ネットワーク１２９を通してＨＴＴＰサーバ１５２へ送る。ウェブ・ページ２１４をロードする前に、下記のイベント、すなわち、（ａ）マスター・アプレット１２４Ａに対する、ブラウザ１１４Ａのストップ・ルーチンの実行指示、（ｂ）ブラウザ１１４Ａおよびウェブ・ページ２０４に対して確立されたソケット接続を通してのマスター・アプレット１２４Ａにより、ウェブ・ページ２０４が、アンロードされたことをＷＴＳサーバ１４４に知らせるためのコマンドの送信と、ブラウザ１１４Ａ、およびウェブ・ページ２０４に対して確立したソケット接続の切り離し、（ｃ）ＷＴＳサーバ１４４によるコマンドを受信した時間を示すタイム・スタンプ（アンロード時間）の発行、および（ｄ）ブラウザ１１４Ａに対して生成されたセッションへのウェブ・ページ２０４のＵＲＬおよびタイム・スタンプの記録が発生する。図６を参照しながら、以下の説明を読めば分かるように、ＵＲＬ、コマンドおよびアンロード時間は、前記セッションに対して生成されたＵＲＬ経歴リストおよびコマンド・リストに記憶される。点線（２）で示すように、ＨＴＴＰサーバ１５２は、消費者ページ貯蔵所１４６からのウェブ・ページ２１４を検索し、それをブラウザ１１４Ａに送る。ウェブ・ページ２０４のように、ウェブ・ページ２１４は、マスター・アプレット１２４Ａ、ＤＴＳアプレット１２６Ａ、およびセッションＩＤアプレット１２８Ａの位置を示すための一組のアプレット・タグ２０８を含む。点線（３）で示すように、ウェブ・ブラウザ１１４Ａは、ウェブ・ページ２１４をロードする。点線（４）で示すように、ウェブ・ページ２１４のロードに応じて、ウェブ・ブラウザ１１４Ａは、ブラウザ１１４Ａによりメモリ領域１１５Ａ内のキャッシュに記憶された（アプレット・タグ２０８の表示に基づいて）、マスター・アプレット１２４Ａ、ＤＴＳアプレット１２６Ａ、およびセッションＩＤアプレット１２８Ａの位置を発見し、前記アプレットを初期化し、その後これらのアプレットを呼び出す。点線（５）で示すように、マスター・アプレット１２４Ａは、専用ソケットを開き、ブラウザ１１４Ａおよびウェブ・ページ２１４に対するＷＴＳゲートウェイ１４２へのソケットの接続を確立する。ブラウザ１１４Ａおよびウェブ・ページ２１４に対して確立したソケット接続を通して、マスター・アプレット１２６Ａは、ＷＴＳサーバ１４４に、ウェブ・ページ２１４がロードされたことを知らせるために、ブラウザ１１４Ａに対して一意のＩＤおよびウェブ・ページ２１４のＵＲＬと一緒に、コマンドを送る。ＷＴＳサーバ１４４は、コマンドを受信した時間を示すタイム・スタンプ（ロード時間）を発行し、ウェブ・ページのＵＲＬおよびタイム・スタンプをブラウザ１１４Ａに対して生成したセッションに記憶する。図６を参照しながら下記説明を読めば分かるように、ＵＲＬ、コマンドおよびロード時間は、セッションに対して生成されたＵＲＬ経歴リストおよびコマンド・リストに記憶される。

【００３５】図５について説明すると、この図は、前記アプレットおよびウェブ・ページ２２４が、ブラウザ１１４Ａにより、端末装置１０４Ａにすでにダウンロードされ、キャッシュに記憶されているとき、本発明の動作を行うために、（ウェブページ２１４の後ろの）以降のウェブ・ページ２２４のロードに応じて、（消費者）マスター・アプレット１２４Ａ、ＤＴＳアプレット１２６Ａ、およびセッションＩＤアプレット１２８Ａを呼び出すプロセスを示す。

【００３６】点線（１）で示すように、ウェブ・ブラウザ１１４Ａは、ブラウザ１１４Ａにより維持されているメモリ領域１１５Ａに、キャッシュに記憶されているウェブ・ページ２２４をロードする。ウェブ・ページ２０４および２１４のように、ウェブ・ページ２２４は、マスター・アプレット１２４Ａ、ＤＴＳアプレット１２６Ａおよびセッションアプレット１２８Ａの位置を示す、一組のアプレット・タグ２０８を含む。ウェブ・ページ２２４をロードする前に、下記のイベント、すなわち、（ａ）マスター・アプレット１２４Ａに対するブラウザ１１４Ａのストップ・ルーチンの実行指示、（ｂ）ブラウザ１１４Ａおよびウェブ・ページ２１４Ａに対して確立されたソケット接続を通してのマスター・アプレ

ット１２４Ａにより、ウェブ・ページ２１４がアンロードされたことをＷＴＳサーバ１４４に知らせるためのコマンドの送信と、ブラウザ１１４Ａおよびウェブページ２１４に対して確立したソケット接続の切り離し、（ｃ）ＷＴＳサーバ１４４によるコマンドを受信した時間を示すタイム・スタンプ（アンロード時間）の発行、および（ｄ）ブラウザ１１４Ａに対して生成されたセッションへのウェブ・ページ２１４のＵＲＬおよびタイム・スタンプの記録が発生する。図６を参照しながら、以下の説明を読めば分かるように、ＵＲＬ、コマンドおよびアンロード時間は、前記セッションに対して生成されたＵＲＬ経歴リストおよびコマンド・リストに記憶される。点線（２）で示すように、ウェブ・ページ２２４のロードに応じて、ブラウザ１１４Ａは、ブラウザ１１４Ａにより端末装置１０４Ａのメモリ領域１１５Ａにキャッシュされたマスター・アプレット１２４Ａ、ＤＴＳアプレット１２６Ａ、およびセッションＩＤアプレット１２８Ａの位置を発見し、前記アプレットを初期化し、呼び出す。点線（３）で示すように、マスター・アプレット１２４Ａは、専用ソケットを開き、ブラウザ１１４Ａおよびウェブ・ページ２２４に対する、ＷＴＳゲートウェイ１４２へのソケットの接続を確立する。ブラウザ１１４Ａおよびウェブ・ページ２２４に対して確立したソケット接続を通して、マスター・アプレット１２６Ａは、ＷＴＳサーバ１４４に、ウェブ・ページ２２４がロードされたことを知らせるために、ブラウザ１１４Ａに対して一意のＩＤ、およびウェブ・ページ２２４のＵＲＬと一緒に、コマンドを送る。ＷＴＳサーバ１４４は、コマンドを受信した時間を示すタイム・スタンプ（ロード時間）を発行し、前記ＵＲＬおよび前記タイム・スタンプをブラウザ１１４に対して生成したセッションに記憶する。図６を参照しながら下記説明を読めば分かるように、ＵＲＬ、コマンドおよびロード時間は、セッションに対して生成されたＵＲＬ経歴リストおよびコマンド・リストに記憶される。

【００３７】図５に示す実施形態においては、ＨＴＴＰサーバ１４４に要求が到着していなくても、ウェブ・ページ２２４が、端末装置１０４Ａのキャッシュ付きメモリからロードされたとき、マスター・アプレット１２４Ａは、ブラウズするためのアクティビティをＷＴＳサーバ１４４に送り続ける。

【００３８】端末装置１０４Ａに対するマスター・アプレット、ＤＴＳアプレットおよびセッションＩＤアプレットのロードおよび呼出の図３から図５に示すプロセスは、端末装置１０４Ｋ、１０４Ｎに対しても使用することができることに留意されたい。

【００３９】図３から図５においては、マスター・アプレット、ＤＴＳアプレット、およびセッションＩＤアプレットは、すべて、ウェブ・ページ２０４、２１４および２２４に埋め込まれる。しかし、ウェブ・ページにすべてのアプレットを埋め込む必要はないことに留意されたい。実行する必要がある機能により、ウェブ・ページにアプレット・タグを選択的に設定することにより、ウェブ・ページに各アプレットを選択的に埋め込むことができる。例えば、個々の素子のデータ同期および追跡が必要ない場合には、ＤＴＳアプレットをリンクするためのアプレット・タグをウェブ・ページから除去することができる。同じトークンにより、追加の機能が必要な場合には、追加のアプレットをリンクするために、追加のアプレット・タグをウェブ・ページに追加することができる。

【００４０】図６について説明すると、この図は、本発明のより詳細なセッション表１４５（図１参照）である。

【００４１】その各端末装置におけるブラウザが、ウェブ・サイト１３４で、ウェブ・ページをブラウズしている間、ＷＴＳサーバ１４４は、すべてのブラウザとウェブ・サイト１３４から前記ブラウザにすでにダウンロードされたウェブ・ページとの間の相互作用についての情報を収集し、分析する。前記情報を収集し、分析する際一つ問題になることは、ウェブ・サイト１３４で、個々のウェブ・ページをブラウズするということは、状態のないプロセスであるということである。より詳細に説明すると、ウェブ・サイト１３４は、異なるブラウザから一連の要求を受信し、前記一連の要求に応じて各ウェブ・ページを各ブラウザに送る。個々のブラウザからの要求を処理する際に、ウェブ・サイト１３４は、１：１の関係を維持するために、同じブラウザと不変の接続を維持するので、ウェブ・サイト１３４は、ブラウザからの一連の要求を制御することもできないし、その上でデータを維持することもできない。

【００４２】ブラウザとウェブ・ページとの間の相互作用に関する情報を意味のあるように収集および分析するために、あるセッションは、特定のブラウザからある時間の間に発生するウェブ・ページの相互作用の収集として定義される。あるブラウザが最初にウェブ・サイト１３４を見つけたときに、あるセッションが生成され、セッション・ウィンドウ（または、セッション長）がそのセッションに対して定義される。（前記ブラウザに対して一意のＩＤにより識別され、各マスター・アプレットにより発行された）特定のブラウザからのアクティビティが、前記セッション・ウィンドウの間に発生しない場合には、ＷＴＳサーバ１４４により、前記セッションは終了し、クリーンアップされる。セッション・ウィンドウは、関連ブラウザに関する情報が、ＷＴＳサーバ１４４に送られる度に、リフレッシュされる（時間ゼロにリセットされる）。例えば、あるセッション・ウィンドウが１５分と定義された場合には、関連端末装置が１５分毎にあるアクティビティをしている限りは、前記セッションは開いた状態に維持される。１５分間アクティビテ

ィが行われない場合には、前記セッションは終了し、除去される。同じ端末装置からの以降の要求があると、新しいセッションが生成される。ある端末装置に対してあるセッションが生成されると、一つまたはそれ以上の端末装置が、そのセッションに参加することができる。

【００４３】図6に示すように、セッション表１４５は、それぞれ、M個のセッションに対して生成されたM個のセッションＩＤを含む。各セッションＩＤは、（１）あるセッションに関連する情報を維持するためのセッション・リスト、（２）あるセッションのすべての参加ブラウザに関する情報を維持するための参加者リスト（注：あるセッションが最初に生成された場合には、そのセッションは一人の参加者しか含んでいない）、（３）あるセッションへのすべての参加者がアクセスした、すべてのウェブ・ページに関する情報を維持するためのＵＲＬ経歴リスト、（４）あるセッション中にすべての参加者がアクセスした、ウェブ・ページ上のデータ・フィールドに関する情報を維持するためのデータ・リスト、および（５）あるセッション中に種々の参加者により、ＷＴＳサーバ１４４に対して発行された、すべてのコマンドに関する情報を維持するためのコマンド・リストに関連する。

【００４４】セッション・リストに含まれる通常の項目は、（１）あるセッションを識別するためのセッションＩＤ、（２）セッションが生成された実際の名前を示すユーザ名、（３）そのセッションのスタート時間を示すスタート時間、（４）前記セッションの停止時間を示す停止時間、および（５）前記セッションの注を記録するためのセッション注がある。

【００４５】参加者リストに含まれる通常のフィールドとしては、（１）あるセッションに、参加者リストをリンクするためのセッションＩＤ、（２）ある参加者を識別するための参加者ＩＤ、（３）参加者のＩＰアドレスを示すための参加者アドレス、（４）参加者のユーザ・クラスを示すためのクラス（顧客、エージェント、スーパーバイザ、アドミニストレータ等）、および（５）参加者ブラウザに対する同期方向を示すための方向がある。

【００４６】ＵＲＬ来歴リストに含まれる通常のフィールドとしては、（１）あるセッションに前記ＵＲＬ来歴リストをリンクするためのセッションＩＤ、（２）アクセスしたウェブ・ページのＵＲＬを示すためのページＵＲＬ、（３）前記ウェブページにアクセスしたある参加者を識別するための参加者ＩＤ、（４）前記ウェブ・ページのロード時間を示すためのロード時間、および（４）前記ウェブ・ページのアンロード時間を示すためのアンロード時間がある。

【００４７】データ・リストに含まれる通常のフィールドとしては、（１）あるセッションにデータ・リストをリンクするためのセッションＩＤ、（２）前記データ・フィールドがすでにブロードキャストされたことを示すための中継済み、（２）前記データ・フィールドの実際の名前を示すフィールド名、（３）ウェブ・ページ上に表示された前記データ・フィールドの名前を示すためのデータ名、（４）前記データ・フィールドの数値を示すためのデータ数値、（５）前記データ・フィールドが更新された時間を示すためのタイム・スタンプ、（６）前記データ・フィールドが表示されたウェブ・ページを示すためＵＲＬ、および（７）前記データ・フィールドを更新した参加者ブラウザを示すための参加者ＩＤがある。

【００４８】コマンド・リストに含まれる通常のフィールドとしては、（１）あるセッションに前記データリストをリンクするためのセッションＩＤ、（２）実行した特定のコマンド（ページのロード、ページのアンロード、データ・フィールドの変更等）を示すためのコマンド、（３）コマンドが実行されたウェブ・ページを示すためのＵＲＬ、（４）コマンドが実行されたデータ・フィールドを示すフィールド点、および（５）コマンドが実行された時間を示すタイム・スタンプがある。

【００４９】セッション表１４５から、あるセッションが除去される前に、データベース処理アプリケーション１４７は、関連セッション・リスト、ＵＲＬ来歴リストおよびコマンド・リストをデータベース１５６に記憶する。前記三つのリストに含まれるデータは、後でデータ倉庫集積アプリケーションにより使用される。

【００５０】図7について説明すると、この図は、本発明の管理ページ貯蔵所１４９からエージェント・ページ（または、スーパーバイザ・ページ）をダウンロードすることにより、あるエージェント・ページ（または、あるスーパーバイザ・ページ）に対する、セッション・インターフェースを生成するための動作である。図7の実施形態の場合には、管理ユーザ・クラス（エージェント・クラス、またはスーパーバイザ・ユーザ・クラス）が、端末装置１０４Ｎに割り当てられているものと仮定する。その結果、ＨＴＴＰサーバ１５２の機密保護アプリケーションが、消費者ページ貯蔵所１４６および管理ページ貯蔵所１４９両方に記憶されたウェブ・ページへのアクセスを許可する。

【００５１】ステップ７０２においては、端末装置１０４Ｎのところのあるエージェントが、端末装置１０４Ｎからあるエージェントの ＵＲＬを入力する、ブラウザ１１４Ｎが、マスター・アプレットおよびエージェント・アプレットが埋め込まれているエージェント・ページを検索するために、ＨＴＴＰサーバ１５２に前記ＵＲＬを送る。スーパーバイザに対しては、前記エージェントは、端末装置１０４Ｎのところで、あるスーパーバイザＵＲＬを入力し、ブラウザ１１４Ｎは、マスター・アプレットおよびスーパーバイザ・アプレットが埋め込まれている、スーパーバイザ・ページを検索するために、ＨＴＴ

Pサーバ152に前記URLを送る。

【0052】ステップ704においては、HTTPサーバ152は、管理ページ貯蔵所149から前記エージェント・ページ（または、スーパーバイザ・ページ）を検索し、それをブラウザ114Nに送る。

【0053】ステップ706においては、ブラウザ114Nは、マスター・アプレット（管理マスター・アプレット）およびエージェント・アプレットが埋め込まれている、前記エージェント・ページをダウンロードするか、または、マスター・アプレット（管理マスター・アプレット）およびスーパーバイザ・アプレットが埋め込まれている前記スーパーバイザ・ページをダウンロードする。

【0054】ステップ708においては、ブラウザ114Nは、HTTPサーバ152からマスター・アプレットおよびエージェント・アプレットをダウンロードし、前記アプレットを初期化し、呼び出すか、または、HTTPサーバ152からマスター・アプレットおよびスーパーバイザ・アプレットをダウンロードし、前記アプレットを初期化し、呼び出す。

【0055】ステップ710においては、マスター・アプレットは、専用ソケットを開き、WTSゲートウェイ142へのソケット接続を確立し、ブラウザ114Nに一意のIDをWTSサーバ144に送る。WTSサーバ144は、前記一意のIDに基づいて、ブラウザ114Nを識別することができる。

【0056】ステップ712においては、エージェント・アプレットが、前記エージェント・ユーザに対して、図8Aに示すエージェント・セッション・インターフェース800Aを生成するか、または、スーパーバイザ・アプレットが、前記スーパーバイザ・エージェントに対して、図8Bに示すスーパーバイザ・セッション・インターフェース800Bを生成する。

【0057】図8Aについて説明すると、この図は、本発明により、ステーション712において、あるエージェントに対して生成されたエージェント・セッション・インターフェース800Aである。

【0058】図8Aに示すように、前記セッション・インターフェースは、セッションIDを入力するためのテキスト・ボックス804、前記セッションIDにより識別された、あるセッションに参加するための参加セッション・ボタン806、あるセッションから離脱するためのドロップ・ボタン808、（同期の際に、リーディング・ブラウザとして、あるブラウザを指名するものを選択する）リーダ・セッション・ボックス810、（同期の際にフォローイング・ブラウザとして、ブラウザを指名するものを選択する）フォロア・チェック・ボックス812、選択したセッションに関連する参加者リストに含まれる情報を表示するためのスクロール可能なリスト・ボックス816、識別したURL来歴リストの情報を表示するためのスクロール可能なリスト・ボックス818、および識別したデータ・リストの情報を表示するためのテキスト・ボックス820を含む。リーダおよびフォロア・チェック・ボックス810および812の両方が、エージェント・セッション・インターフェースで選択された場合には、ブラウザ114Aは、同期中、リーディングおよびフォローイング・ブラウザとしての働きをする。

【0059】図8Bについて説明すると、この図は、本発明により、ステップ712において、スーパーバイザに対して生成された、スーパーバイザ・セッション・インターフェース800Bを示す。

【0060】図8Bに示すように、前記セッション・インターフェースは、セッション表145のすべての能動セッションのセッションIDを表示するための、またセッションIDの中の一つを選択するためのスクロール可能なリスト・ボックス832、WTSサーバ144の関連統計を表示し、スクロール可能なリスト・ボックス832で選択したセッションについての詳細を表示するためのテキスト・ボックス834、スクロール可能なリスト・ボックス832で選択したセッションに関する詳細を表示するための複数のコラムからなるスクロール可能なリスト・ボックス836、スクロール可能なリスト・ボックス832からあるセッションを選択するためのセッション選択ボタン838を含む。スクロール可能なリスト・ボックス832内の情報を使用することにより、スーパーバイザは、すべての能動セッションを監視することができる。複数のコラムからなるスクロール可能なリスト・ボックス836内の情報を使用することにより、スーパーバイザは、（1）このセッションがあるエージェントの協力を得ているかどうか、（2）ユーザ名、および（3）エージェントIDを含む、スクロール可能なリスト・ボックス832から選択したセッションの動作状態を監視することができる。選択セッション・ボタン838を選択することによって、スーパーバイザは、図8Cに示すように、エージェント・セッション・インターフェースを生成することができる。

【0061】図8Cについて説明すると、この図は、本発明のスーパーバイザ・エージェント・セッション・インターフェース800Cである。

【0062】図9について説明すると、この図は、本発明による、あるエージェントがあるセッションに参加するための動作を示すフローチャートである。

【0063】図9の実施形態の場合には、（1）端末装置104Aのところの消費者は、ブラウザ114Aを通して消費者ページ貯蔵所146からウェブ・ページをブラウズしている、（2）図6のセッション・リスト1は、ブラウザ114Aに対して生成されたものである、（3）ブラウザ114Nには、あるエージェント・クラスがすでに割り当てられている、（4）図8Aのエージ

ェント・セッション・インターフェース８００Ａは、端末装置１０４Ｎ上にすでに表示されている、（５）（管理）マスター・アプレットおよびエージェント・アプレットは、ブラウザ１１４Ｎにすでにダウンロードされている、（６）専用ソケット接続は、（管理）マスター・アプレットにより、端末装置１０４Ｎのところに表示されているセッション・インターフェース８００Ａに対して確立されている、および（７）端末装置１０４Ａのところのエージェントは、呼出センターで作業を行っているものと仮定する。

【００６４】図９に示すように、ステップ９０２においては、消費者は、端末装置１０４Ａのところでウェブ・ページをブラウズしている。前記ウェブ・ページにおいては、セッションＩＤアプレット１２８Ａは現在のセッションＩＤを表示する。消費者が呼び出すことができる呼出センターの電話番号も、前記ウェブ・ページ上に表示される。

【００６５】ステップ９０４においては、消費者は、電話102Ａにより、前記電話番号をダイヤルすることにより前記呼出センターに接続され（図１参照）、前記呼出は、前記呼出センターにより、前記エージェントに接続される。

【００６６】ステップ９０６においては、電話１０２Ａを通して、消費者は、前記エージェントに表示されている現在のセッションＩＤを知らせる。電話番号を使用する代わりに、前記エージェントは、他の方法により、現在のセッションＩＤを知ることができることに留意されたい。例えば、消費者は、現在のセッションＩＤと一緒に、消費者の呼出ＩＤを含む、特別のウェブ・ページに自分の電話番号を記載することができる。前記情報は、現在のセッションＩＤを参照するために、エージェントが使用することができる特殊な参照表内に記憶することができる。

【００６７】ステップ９０８においては、端末装置１０４Ｎにおいて、前記エージェントが、テキスト・ボックス８０４に現在のセッションＩＤを入力する（図８Ａ参照）。

【００６８】ステップ９１０においては、焦点が合わなくなったことに応じて、またはエンター・キーを押すことによって、端末装置１０４Ｎ上に表示されている、エージェント・セッション・インターフェース８００Ａに対して確立されたソケット接続を通して、端末装置１０４Ｎのところの（管理）マスター・アプレットが、エージェント・アプレットに対する、参加者リスト１、ＵＲＬ来歴リスト１、およびデータ・リスト１の情報を検索するために、ＷＴＳサーバ１４４にコマンドを送る。

【００６９】ステップ９１２においては、ＷＴＳサーバ１４４は、要求された情報を（マスター・アプレットを通して）エージェント・アプレットに送る。

【００７０】ステップ９１４においては、端末装置１０４Ｎのところのエージェント・アプレットは、それぞれ、（参加者）スクロール可能なリスト・ボックス８１６、および（ＵＲＬ来歴）スクロール可能なリスト・ボックス８１８内に、参加者リスト１およびＵＲＬ来歴リスト１からのある種の情報を表示する。

【００７１】ステップ９１６においては、エージェントは、端末装置１０４Ｎ上に表示されている、エージェント・セッション・インターフェース８００Ａの参加ボタン８０６を選択する。

【００７２】ステップ９１８においては、ステップ９１６における選択に応じて、端末装置１０４Ｎ上に表示された、エージェント・セッション・インターフェースのために確立されたソケット接続を通して、（管理）マスター・アプレットは、選択したセッションに参加するように、ＷＴＳサーバ１４４にコマンドを送る。前記ソケット接続に関連する識別に基づいて、ＷＴＳサーバは、ブラウザ１１４Ｎに対する参加者ＩＤを発生し、端末装置１０４Ｎに対する参加者アドレスを発見することができる。

【００７３】ステップ９２０においては、ＷＴＳサーバ１４４は、前記参加者ＩＤおよび前記参加者アドレスを参加者リスト１に記憶する。このステップにおいて、参加者リスト１は、それぞれ、ブラウザ１１４Ａおよび１１４Ｎに対する参加者ＩＤを含む、二つの参加者記録（二つの横列）を含む。

【００７４】ステップ９２２においては、端末装置１０４Ｎのところで、エージェントはリーディング・チェック・ボックス８１０またはフォローイング・チェック・ボックス８１２、または両方を選択する。リーダ・チェック・ボックス８１０だけを選択することにより、端末装置１０４Ｎのところでのアクティビティは、端末装置１０４Ａのところで同期されるが、他の方法は使用できない。フォロア・チェック・ボックス８１２だけを選択することにより、端末装置１０４Ａのところでのアクティビティは、端末装置１０４Ｎのところで同期されるが、他の方法は使用することはできない。リーダ・チェック・ボックス８１０およびフォロアチェック・ボックス８１２の両方を選択することによって、端末装置１０４Ａおよび１０４Ｎのところのアクティビティは、相互に同期（両方向同期）する。この選択に応じて、エージェント・セッション・インターフェース８００Ａのために確立されたソケット接続を通して、（管理）マスター・アプレットは、ＷＴＳサーバ１４４に、前記同期方向を示すコマンドを送る。ＷＴＳサーバ１４４は、参加者リスト１の二つの記録の方向フィールドに、前記同期方向情報を記憶する。この実施形態の場合には、両方向同期は端末装置１０４Ａおよび１０４Ｎに対して選択されたものと仮定する。

【００７５】ステップ９２４においては、ＷＴＳサーバ１４４は、（管理）マスター・アプレットに、端末装置

104Aのところで現在ブラウズされている、ウェブ・ページのURLを送る。

【0076】ステップ926においては、端末装置104Nのところで、エージェント・アプレットは、図10に示すように、ブラウザ窓1004（第二のブラウザ例）を開く。

【0077】ステップ928においては、ブラウザ114Nは、消費者ページ貯蔵所146からURLにより識別したウェブ・ページをダウンロードし、それをブラウザ窓1004内に表示する。（消費者）マスター・アプレット、一組のDTSアプレット、およびセッションIDアプレットは、ダウンロードされたウェブ・ページに埋め込まれる。

【0078】ステップ930においては、ブラウザ114Nは、（消費者）マスター・アプレット124N、一組のDTSアプレット126N、およびセッションIDアプレット128Nをダウンロードする。

【0079】ステップ932においては、端末装置104Nのところの第二のブラウザ窓1004に表示されているウェブ・ページが、端末装置104Aのところに表示されているウェブ・ページと同期する。

【0080】ステップ932の後で、端末装置104Nのところのエージェント（第一のエージェント）が、他のエージェント（第二のエージェント）からの助力を必要とする場合には、第一のエージェントは、第二のエージェントを呼び出して、自分の現在のセッションIDを知らせることができる。その後、第二のエージェントは、図8Aに示すように、端末装置104Kのところに表示されている、エージェント・セッション・インターフェースを使用して、現在のセッションに参加することができる。

【0081】図10について説明すると、この図は、本発明の端末装置104Nのところの二つのブラウザ例（800Aおよび1004）を含む、スクリーン・ディスプレイである。

【0082】図10に示すように、端末装置104Nのところで、第一のブラウザ例は、現在のセッションを制御し、監視するために、エージェント・セッション・インターフェース800Aを供給し、エージェント・セッション・インターフェース800A用の（管理）マスター・アプレットは、エージェント・セッション・インターフェース800Aのためのソケット接続を確立し、維持する。第二のブラウザ例は、同期しているウェブ・ページを表示するために、ブラウザ窓1004を供給する。（消費者）マスター・アプレット124Nは、ブラウザ窓1004に表示されている各ウェブ・ページに対するソケット接続を確立し、維持する。

【0083】図11について説明すると、この図は、本発明のウェブ・ページ同期の動作を示すフローチャートである。

【0084】図11の実施形態の場合には、（1）端末装置104Aのところの消費者は、ブラウザ114Aを通して消費者ページ貯蔵所146からウェブ・ページをブラウズし、（2）ブラウザ114Aに対してあるセッションが生成され、（3）図6のセッション・リスト1および参加者リスト1は、前記セッションに対してすでに生成されていて、（4）端末装置104Aおよびすべての参加者端末装置に対して、両方向同期がすでに選択されていて、（5）（消費者）マスター・アプレット、DTSアプレット、およびセッションIDアプレットが、ブラウザ104Aおよびすべての参加者ブラウザに、すでにダウンロードされているものと仮定する。

【0085】図11に示すように、ステップ1104においては、ブラウザ114Aは、消費者ページ貯蔵所146、または端末装置104Aのメモリ領域115Aからウェブ・ページをロードする。マスター・アプレット124A、DTSアプレット126A、およびセッションIDアプレット128Aが、まだブラウザ114Aにダウンロードされていない場合には、ブラウザ114Aは、前記アプレットを消費者ページ貯蔵所146からダウンロードする。しかし、この実施形態の場合には、前記アプレットは、ダウンロードされるものと仮定する。

【0086】ステップ1106においては　ウェブ・ページのロードに応じて、ブラウザ114Aは、マスター・アプレット124A、DTSアプレット126A、およびセッションIDアプレット128Aを初期化し、呼び出す。

【0087】ステップ1108においては、マスター・アプレット124Aは、（1）専用ソケットを開き、ブラウザ114Aおよびロードされたウェブ・ページに対する、WTSゲートウェイ142へのソケット接続を確立し、および（2）前記ソケット接続を通して、WTSサーバ144は、ブラウザ114Aに対して一意のID、およびロードしたウェブ・ページのURLと一緒にコマンドを送る。前記一意のコマンドに基づいて、WTSサーバは、ブラウザ114Aに対して生成されたセッションを識別することができる。

【0088】ステップ1110においては、WTSサーバ144は、ブラウザ114Aに対するセッションを識別する。

【0089】ステップ1112においては、WTSサーバ144は、（図6に示すように）参加者リスト1の参加者端末装置に割り当てられたすべてのIPアドレスの位置を発見し、すべての参加者端末装置に、URLと一緒にコマンドをおくる。（ただし、WTSサーバ144は、端末装置104AにはURLを送らない。何故なら、前記URLは端末装置104Aからのものであるからである。）ステップ1114においては、前記コマンドを受け取ったとき、参加者端末装置の（消費者）マスター・アプレットは、自らを初期化し、URLをその各

ブラウザに送る。

【0090】ステップ1116においては、参加者端末装置の各ブラウザは、URLに従ってウェブ・ページをダウンロードし、表示する。端末装置104Aのように、（エージェント・セッション・インターフェースが表示されている）各参加者端末装置は、図11に示す動作を使用して、ページ同期をリードすることができる。

【0091】図12Aについて説明すると、この図は、五つのデータ・フィールド、すなわち、名前1202、期間1204、残高1206、支払い1208、コメント1210、現在のセッションIDを表示するためのテキスト・ボックス1212、および消費者が呼び出すことができる、呼出センター番号を表示するためのテキスト・ボックスを含むウェブ・ページである。

【0092】図12Bについて説明すると、この図は、名前フィールド202のデータが、スーザン・キングからスー・グラントに変更され、本発明によりこの変更が参加者端末装置のところで同期している点以外は、図12A類似のウェブ・ページである。

【0093】図13について説明すると、この図は、本発明のデータ同期の動作を示すフローチャートである。

【0094】図13に示す実施形態の場合には、（1）端末装置104Aのところの顧客は、ブラウザ114Aを通してウェブ・ページをブラウズ中であり、（2）端末装置104Aに対してあるセッションが生成され、（3）図6に示すセッション・リスト1および参加者リスト1は、前記セッションに対してすでに生成されていて、（4）端末装置104Nが参加者の一人であり、（5）図12Aに示す五つのデータ・フィールドを含むウェブ・ページ1200が、端末装置104Aおよびすべての参加者端末装置上に表示され、（6）端末装置104Aおよびすべての参加者端末装置に対して、両方向同期がすでに選択されていて、（7）（消費者）マスター・アプレット、DTSアプレット、およびセッションIDアプレットが、ブラウザ114Aおよびすべての参加者ブラウザに、すでにダウンロードされていて、（8）前記DTSアプレットが、五つの個々のDTSアプレット、すなわち、DTSアプレット1、DTSアプレット2、DTSアプレット3、DTSアプレット4、およびDTSアプレット5を含み、（9）前記五つの個々のDTSアプレットが、図12Aに示すウェブ・ページ1200の五つのデータ・フィールド上で発生したイベントの監視および処理に責任を持ち、（10）（消費者）マスター・アプレット124Aは、端末装置104Aに表示されているウェブ・ページ12Aに対して、WSTゲートウェイ142への専用ソケット接続をすでに確立し、（11）端末装置104Aのところの消費者が、名前フィールド1202をスーザン・キングからスー・グラントに変更したいと考えているものと仮定する。

【0095】図13に示すように、ステップ1304においては、顧客は、名前フィールド1202をスーザン・キングからスー・グラントに変更する。

【0096】ステップ1306においては、名前フィールド1202上の焦点がぼけるか、エンター・キーが押されたことに応じて、DTSアプレット1は前記変更を検出し、前記変更をマスター・アプレット124Aに送る。

【0097】ステップ1308においては、専用ソケット接続を通して、マスター・アプレット124Aは、WTSサーバ144に名前フィールド1202の変更と一緒に、コマンドを送る。前記変更が、ウェブ・ページ1200に対して確立された前記専用ソケット接続を通してWTSサーバ144に送られるので、WTSサーバ144は、コマンド、ウェブ・ページ1200、およびその上で変更が行われた名前フィールドの起源を認識することができる。

【0098】ステップ1310においては、WTSサーバ144は、ブラウザ114Aに対して生成されたセッションを識別する。

【0099】ステップ1312においては、WTSサーバ144は、参加者リスト1の参加者ブラウザに割り当てられたIPアドレスの位置を発見し、（名前フィールド1202の変更と一緒に）すべての参加者端末装置のマスター・アプレットへコマンドをおくる。（ただし、WTSサーバ144は、ブラウザ114Aには前記コマンドおよび変更を送らない。何故なら、前記変更はブラウザ114Aからのものであるからである。）ステップ1314においては、前記コマンドを受け取ったとき、（マスター・アプレット124Nを含む）（消費者）マスター・アプレットは、ブラウザ114NにおけるDTSアプレット1を含む、その各DTSアプレットに名前フィールド1200の変更を送る。

【0100】ステップ1316においては、DTSアプレット1は、端末装置104Nを含む各端末装置上の表示されている、各ウェブ・ページ1200上の名前フィールドに、「スーザン・グラント」を表示し、更新する。

【0101】図13に示す動作は、図12Aに示すウェブ・ページ1200上の他の四つのデータ・フィールドに対して、データ同期を行うために使用される。

【0102】データ・フィールド同期は、端末装置104Nにおいても行うことができることに留意されたい。例えば、図12Bに示すように、端末装置104Nにおけるエージェントが、「口座名は、変更されました」というコメントをウェブページ1200'上のコメント・フィールド1210'に入力した場合には、この更新は、図13に示す動作を使用することにより、端末装置104Aのところのコメント・フィールド1210に表示される。

【0103】図14について説明すると、この図は、本発明のウェブ・ページ追跡の動作を示すフローチャートである。

【0104】図14の実施形態の場合には、（1）端末装置104Aのところの顧客は、ブラウザ114Aを通してウェブ・ページをブラウズ中であり、（2）端末装置104Aおよびすべての参加者端末装置に対してあるセッションが生成され、（3）図6に示すセッション・リスト1および参加者リスト1、およびはURL来歴リスト1が、前記セッションに対してすでに生成されていて、（4）端末装置104Aおよびすべての参加者端末装置に対して、両方向同期がすでに選択されていて、（5）（消費者）マスター・アプレット、DTSアプレット、およびセッションIDアプレットが、端末装置104Aおよびすべての参加者端末装置に、すでにダウンロードされているものと仮定する。

【0105】図14に示すように、ステップ1404においては、ブラウザ114Aは、消費者ページ貯蔵所146、または端末装置104Aのメモリ領域115Aからウェブ・ページをダウンロードする。マスター・アプレット124A、DTSアプレット126A、およびセッションIDアプレット128Aが、まだ端末装置104Aにダウンロードされていない場合には、ブラウザ114Aは、前記アプレットをHTTPサーバ152からダウンロードする。しかし、この実施形態の場合には、前記アプレットは、すでにダウンロードされている。

【0106】ステップ1406においては、ブラウザ114Aは、マスター・アプレット124A、DTSアプレット126A、およびセッションIDアプレット128Aを初期化し、呼び出す。

【0107】ステップ1408においては、マスター・アプレット124Aは、専用ソケットを開き、ブラウザ114Aおよびロードされたウェブ・ページに対する、WTSゲートウェイ142へのソケット接続を確立する。その後、マスター・アプレット124Aは、（1）ブラウザ114Aに対する一意のID、および（2）ロードしたウェブ・ページのURLと一緒に、コマンドを送る。前記ソケット接続を通してコマンドおよびURLを受け取ると、WTSサーバ144は、前記コマンドおよびURLの起源を認識することができる。

【0108】ステップ1410においては、WTSサーバ144は、ブラウザ114Aに対するセッションIDを識別する。

【0109】ステップ1412においては、WTSサーバ144は、セッション・リスト1およびURL来歴リスト1の位置を発見する。

【0110】ステップ1414においては、WTSサーバ144は、前記コマンドを受信した時間（ロード時間）を示すためのタイム・スタンプを発行し、URL来歴リスト1に前記URLおよびタイム・スタンプを記憶する。

【0111】ステップ1416においては、ブラウザ114Aは、WTSサーバ144に、以降のウェブ・ページをロードするようにとの要求を送る。

【0112】ステップ1418においては、前記以降のウェブ・ページをロードする前に、ソケット接続を通して、マスター・アプレット124Aは、WTSサーバ144に、現在のウェブ・ページがすでにアンロードされていることを知らせるために、WTSサーバ144に、URLと一緒にコマンドを送る。

【0113】ステップ1420においては、WTSサーバ144は、端末装置104Aに対するセッションを識別する。

【0114】ステップ1422においては、WTSサーバ144は、セッション・リスト1およびURL来歴リスト1の位置を発見する。

【0115】ステップ1424においては、WTSサーバ144は、前記コマンドを受信した時間を示すためのタイム・スタンプ（アンロード時間）を発行し、URL来歴リスト1に前記URLおよびタイム・スタンプを記憶する。

【0116】ステップ1426においては、マスター・アプレット124Aは、すでにアンロードされたウェブ・ページに対するソケット接続を切り離す。

【0117】図15について説明すると、この図は、本発明のデータ追跡動作を示すフローチャートである。

【0118】図15に示す実施形態の場合には、（1）端末装置104Aのところの顧客は、ブラウザ114Aを通してウェブ・ページをブラウズ中であり、（2）端末装置104Aに対してあるセッションがすで生成され、（3）図6に示すセッション・リスト1および参加者リスト1が、前記セッションに対してすでに生成されていて、（4）端末装置104Nが参加者の一つであり、（5）図12Aに示す五つのデータ・フィールドを含むウェブ・ページ1200が、端末装置104Aおよびすべての参加者端末装置上に表示され、（6）端末装置104Aおよびすべての参加者端末装置に対して、両方向同期がすでに選択されていて、（7）（消費者）マスター・アプレット、DTSアプレット、およびセッションIDアプレットが、端末装置104Aおよびすべての参加者端末装置に、ダウンロードされ、（8）前記DTSアプレットが、五つの個々のDTSアプレット、すなわち、DTSアプレット1、DTSアプレット2、DTSアプレット3、DTSアプレット4、およびDTSアプレット5を含み、（9）前記五つの個々のDTSアプレットが、図12Aに示すウェブ・ページ1200の五つのデータ・フィールド上で発生したイベントの表示、監視および処理の責任を持ち、（10）マスター・アプレット124Aが、端末装置104Aに表示されているウェブ・ページ1200に対して、WTSサーバ1

44への専用ソケット接続をすでに確立し、（11）端末装置104Aのところの顧客が、名前フィールド1202をスーザン・キングからスー・グラントに変更したいと考えているものと仮定する。

【0119】図15に示すように、ステップ1504においては、顧客は、名前フィールド1502をスーザン・キングからスー・グラントに変更する。

【0120】ステップ1506においては、名前フィールド1202上の焦点がぼけるか、エンター・キーが押されたことに応じて、DTSアプレット1は前記変更を検出し、前記変更をマスター・アプレット124Aに送る。

【0121】ステップ1508においては、専用ソケット接続を通して、マスター・アプレット124Aは、WTSサーバ144に名前フィールド1202の変更と一緒に、コマンドを送る。前記変更が、ウェブ・ページ1200に対して確立された、前記専用ソケット接続を通してWTSサーバ144に送られるので、WTSサーバ144は、コマンド、ウェブ・ページ1200、およびその上で変更が行われた名前フィールドの起源を認識することができる。

【0122】ステップ1510においては、WTSサーバ144は、端末装置104Aに対して生成されたセッションを識別する。

【0123】ステップ1512においては、WTSサーバ144は、データ・リスト1に前記URLを記憶し、名前フィールド1202を更新する。

【0124】図15に示した動作は、ウェブ・ページ1200上の他の四つのデータ・フィールドに対するデータ追跡を行う際にも、すべての参加者端末装置に対するデータ追跡を行う際にも使用することができることに留意されたい。

【0125】図16について説明すると、この図は、本発明のスーパーバイザによるセッションへの参加動作を示すフローチャートである。図16に示す実施形態の場合には、スーパーバイザが、呼出センターの端末装置104Kのところで、仕事をしているものと仮定する。

【0126】図16に示すように、ステップ1604においては、スーパーバイザは、図7に示すステップを行う。この場合、スーパーバイザは（マスター・アプレット1と呼ばれる）（管理）マスター・アプレットおよび、スーパーバイザ・アプレットが埋め込まれているスーパーバイザ・ページをダウンロードする。スーパーバイザ・アプレットは、端末装置104K上に、第一のブラウザ例（図17の800B参照）の（図8Bに示すような）スーパーバイザ・セッション・インターフェースを表示する。マスター・アプレット1は、第一のブラウザ例（図17の800B参照）に対する、WTSゲートウェイ142への専用ソケット接続を維持する。

【0127】ステップ1606においては、前記第一のブラウザ例（図17の800B参照）から、スーパーバイザが、（テキスト・ボックス832にリストの形で含まれている）セッションIDを選択し、その後、セッション・ボタン838を選択する。

【0128】ステップ1608においては、選択セッション・ボタン838の選択に応じて、ブラウザ114Kが、（マスター・アプレット2と呼ばれる）（管理）マスター・アプレットおよび、エージェント・アプレットが埋め込まれている、エージェント・ページをダウンロードする。前記エージェント・アプレットは、（図8Cに示すように）スーパーバイザ・エージェント・セッション・インターフェース800Cを生成し、端末装置104K上の第二のブラウザ例（図17の800C参照）にそれを表示する。マスター・アプレット2は、第二のブラウザ例（図17のスーパーバイザ・エージェント・セッション・インターフェース800C参照）に対する、WTSゲートウェイ142への専用ソケット接続を開き、維持する。

【0129】ステップ1610においては、可能なシナリオが二つある。第一のシナリオは、消費者を助けるために、エージェントが選択したセッションに参加し、スーパーバイザが参加者として、前記選択したセッションに参加を希望する。このシナリオの場合、スーパーバイザは、単に参加者ボタン846お選択し、リーダ・ボタン850およびおよびフォロア・ボタン852の両方が自動的に選択される。第二のシナリオは、エージェントが前記セッションに参加せず、スーパーバイザが前記セッションへの参加を希望する。このシナリオの場合、スーパーバイザはエージェントと全く同じように、最初、リーダ・ボタン850および／またはフォロア・ボタン852を選し、その後参加セッション・ボタン846を押すことによりセッションに参加する。この実施形態の場合には、端末装置104Aのところの消費者および端末装置104Nのところのエージェントが、セッションに参加し、スーパーバイザが選択したセッションの参加を希望する。

【0130】ステップ1612においては、スーパーバイザは参加セッション・ボタン846を選択する。

【0131】ステップ1614においては、第二のブラウザ例（図17のスーパーバイザ・エージェント・セッション・インターフェース800C参照）に対するソケット接続を通して、マスター・アプレット2は、WTSサーバ144に、選択したセッションに対する選択したセッションIDと一緒に、コマンドを送る。

【0132】ステップ1615においては、WTSサーバ144は、選択したセッションIDが表示するセッションの位置を発見し、参照番号リスト1およびURL来歴リスト1（図6参照）に記憶している情報をマスター・アプレット2に送る。

【0133】ステップ1616においては、WTSサー

バ144は、ブラウザ114Kに対する参加者リスト1に、参加者IDおよび参加者アドレスを記憶する。このステップにおいては、参加者リスト1は、それぞれ、ブラウザ114A、114Kおよび114Nに対する三つの参加者記録（三つの横列）を含む。

【0134】ステップ1618においては、端末装置104Kのところのエージェント・アプレットは、それぞれ、参加者リスト1に記憶している情報、参加者テキスト・ボックス856内のURL来歴リスト1、およびURL来歴テキスト・ボックス858（図8C参照）を表示する。

【0135】ステップ1620においては、WTSサーバ144は、マスター・アプレット2に、端末装置104Aおよび104Nのところで現在表示されている、ウェブ・ページのURLを送る。

【0136】ステップ1622においては、端末装置104Kところのエージェント・アプレットが、第三のブラウザ例（図17の1704参照）を開く。

【0137】ステップ1624においては、ブラウザ114Kは、消費者ページ貯蔵所146からURLにより識別されたウェブ・ページをダウンロードし（または、そこにキャッシュされている場合には、端末装置104Kのメモリ領域115Kから前記ウェブ・ページをロードし）、それを第三のブラウザ例（図17の1704参照）に表示する。

【0138】ステップ1626においては、ブラウザ114Kは、（前記アプレットがまだダウンロードされていないと仮定して）現在のウェブ・ページにアプレット・タグに従って、消費者ページ貯蔵所146から（消費者）マスター・アプレット124K、DTSアプレット126K、およびセッションIDアプレット128をダウンロードする。

【0139】ステップ1628においては、マスター・アプレット124Kが、図17に示す第三のブラウザ例1704に対する、WTSゲートウェイ142へのソケット接続を開き、確立する。

【0140】ステップ1630の後で、端末装置104Kのところの第三のブラウザ例1704に表示されているウェブ・ページが、端末装置104Aおよび104Nのところに表示されているウェブ・ページと同期する。

【0141】図17について説明すると、この図は、本発明の図16に示すステップに応じる、スーパーバイザに対する三つのブラウザ例（800B、800Cおよび1704）である。

【0142】図18について説明すると、この図は、本発明のあるセッションですでに検討したウェブ・ページの再度のブラウズ動作を示すフローチャートである。

【0143】図18に示す実施形態の場合には、（1）端末装置104Aのところの顧客は、ブラウザ114Aを通して消費者ページ貯蔵所146からのウェブ・ページをブラウズ中であり、（2）ブラウザ114Aに対して、図6に示すセッション・リスト1がすでに生成され、（3）あるエージェント（または、あるスーパーバイザ）が、呼出センターの端末装置104Nのところで仕事をしていて、またエージェント・クラス（または、スーパーバイザ・クラス）が、ブラウザ104Nに割り当てられていて、（4）ブラウザ114Nのところで、図10に示す前記エージェントに対する第一および第二のブラウザ例（または、図17に示すスーパーバイザに対する第一、第二および第三のブラウザ例）が、すでに表示されていて、（5）そのマスター・アプレットにより確立された、その各ソケット接続を通して、図10に示すように、エージェントに対する第一および第二のブラウザ例（または、図17に示すように、スーパーバイザに対する第一、第二および第三のブラウザ例）が、すでにWTSゲートウェイ142に接続され、（6）マスター・アプレット（124Aおよび124N）、DTSアプレット（126Aおよび126N）、およびセッションIDアプレット（128Aおよび128N）が、それぞれ、すでに端末装置104Aおよび104Nにダウンロードされ、（7）エージェント（または、スーパーバイザ）が、すでにブラウザ114Aに対して生成されたセッションを選択し、参加していて、（8）ブラウザ114Nにおいて、図10に示すようなエージェントに対する第二のブラウザ例（または、図17に示すような、スーパーバイザに対する第三のブラウザ例）がブラウザ114Aと同期していて、（9）両方向同期が、ブラウザ114Aおよび114Nに対してすでに選択されていると仮定する。

【0144】ステップ1802においては、エージェント・ユーザに対して、図10に示すエージェント・セッション・インターフェース上のスクロール可能なリスト・ボックス818を通して、エージェント・ユーザは、前記選択したセッションで、ブラウザ114Aにより前にブラウズされた、すべてのウェブ・ページに対するURLを検討する。スーパーバイザの場合には、図17に示すスーパーバイザ・セッション・インターフェース上のスクロール可能なリスト・ボックス858を通して、スーパーバイザは、選択したセッションで、ブラウザ114Aによりすでにブラウズされたすべてのウェブ・ページに対するURLを検討する。

【0145】ステップ1804においては、ブラウザ114Aによりすでにブラウズされた、個々のウェブ・ページを表示するために、エージェント（または、スーパーバイザ）は、スクロール可能なリスト・ボックス818（または、スクロール可能なリスト・ボックス858）からあるURLを選択し、それをダブル・クリックする。

【0146】ステップ1806においては、エージェントの場合、（エージェント）マスター・アプレットまた

は（スーパーバイザ・マスター・アプレット）は、WTSサーバ144に、その各ソケット接続を通して選択したURLと一緒にコマンドを送る。

【0147】ステップ1808においては、WTSサーバ144は、マスター・アプレット124Aおよび124Nに、URLおよび時間情報（ロードおよびアンロード）と一緒にコマンドを送る。その結果、マスター・アプレット124Aおよび124Nは、前記URLに基づいて、ウェブ・ページをロードし、表示するように、その各ブラウザ114Aおよび114Nに知らせることができる。

【0148】ステップ1810においては、WTSサーバ144は、ブラウザ114Aが、URL来歴リスト1およびデータ・リスト1に記憶した情報に基づいて、URLにより識別したウェブ・ページ上で、データ・フィールドに対して、ずでに何等かのアクティビティをしたかどうかをチェックする。図6に示すように、URL来歴リスト1は、（a）ブラウザ114Aの参加者ID、（b）URL、および（c）URLが識別したウェブ・ページのロードおよびアンロード時間に関する情報を含む。データ・リスト1は、（a）データ・フィールドに対するデータ・フィールド名、（b）データ・フィールドの数値、（c）データ・フィールドの数値が変更された時間に関する情報を含む。

【0149】ステップ1812においては、ブラウザ114が、URLにより識別したウェブ・ページ上で、データ・フィールドに対して、ずでに何等かのアクティビティをした場合には、WTSサーバ144は、（データ・フィールド名、データ・フィールドの数値、および時間情報と一緒に）（ブラウザ114Aのところの）マスター・アプレット124A、および（ブラウザ114Nのところの）マスター・アプレット124Nにコマンドを送る。

【0150】ステップ1814においては、ブラウザ114Aのところで、マスター・アプレット124Aは、DSTアプレット126Aに、コマンド、データ・フィールド名、およびデータ・フィールド数値を送る。その結果、DTSアプレット124Aは、現在表示されているウェブ・ページ上の各データ・フィールドに、データ・フィールド値を表示することができる。ブラウザ114Nのところで、マスター・アプレット124Nは、DSTアプレット126Nに、コマンド、データ・フィールド名、およびデータ・フィールド数値を送る。その結果、DTSアプレット124Nは、現在表示されているウェブ・ページ上の各データ・フィールドに、データ・フィールド値を表示することができる。

【0151】URLのロードおよびアンロード時間およびデータ・フィールドに対する設定時間は、URL来歴リスト1およびデータ・リスト1に記録されているので、そうしたい場合には、URLにより識別されたウェブ・ページおよびデータ・フィールドに対して行われたアクティビティを時間情報に従って、コピーすることができる（ウェブ・ページのロード、ウェブ・ページ上でのデータ・フィールドの設定、およびウェブ・ページのアンロード）。

【0152】図19について説明すると、この図は、本発明のあるセッション中にすでに検討した、すべてのウェブ・ページの再ブラウズ動作を示すフローチャートである。

【0153】図19に示す実施形態の場合には、（1）端末装置104Aのところの消費者は、ブラウザ114Aを通して消費者ページ貯蔵所146からのウェブ・ページをブラウズ中であり、（2）ブラウザ114Aに対して、図6に示すセッション・リスト1がすでに生成され、（3）あるエージェント（または、あるスーパーバイザ）が、呼出センターの端末装置104Nのところで仕事をしていて、またエージェント・クラス（または、スーパーバイザ・クラス）が、ブラウザ104Nに割り当てられていて、（4）ブラウザ114Nのところで、図10に示すような、前記エージェントに対する第一および第二のブラウザ例（または、図17に示すスーパーバイザに対する第一、第二および第三のブラウザ例）が、すでに表示されていて、（5）そのマスター・アプレットにより確立された、その各ソケット接続を通して、図10に示すように、エージェントに対する第一および第二のブラウザ例（または、図17に示すように、スーパーバイザに対する第一、第二および第三のブラウザ例）が、すでにWTSゲートウェイ142に接続され、（6）マスター・アプレット（124Aおよび124N）、DTSアプレット（126Aおよび126N）、およびセッションIDアプレット（128Aおよび128N）が、それぞれ、すでに端末装置104Aおよび104Nにダウンロードされ、（7）エージェント（または、スーパーバイザ）が、すでにブラウザ114Aに対して生成されたセッションを選択し、参加していて、（8）ブラウザ114Nにおいて、図10に示すように、前記エージェントに対する第二のブラウザ例（または、図17に示すようなスーパーバイザに対する第三のブラウザ例）が、ブラウザ114Aと同期していて、（9）両方向同期が、ブラウザ114Aおよび114Nに対してすでに選択されていると仮定する。

【0154】ステップ1902においては、エージェント・ユーザの場合には、図10に示すエージェント・セッション・インターフェース上のスクロール可能なリスト・ボックス818を通して、エージェント・ユーザは、前記選択したセッションで、ブラウザ114Aにより前にブラウズされたすべてのウェブ・ページに対するURLを検討する。スーパーバイザの場合には、図17に示すスーパーバイザ・セッション・インターフェース上のスクロール可能なリスト・ボックス858を通し

て、スーパーバイザは、選択したセッションで、ブラウザ１１４Ａによりすでにブラウズされたすべてのウェブ・ページに対するＵＲＬを検討する。

【０１５５】ステップ１９０４においては、ブラウザ１１４Ａにより、すでにブラウズされたすべてのウェブ・ページを表示するために、エージェント（または、スーパーバイザ）は、図１０に示すように、エージェント・セッション・インターフェースの「ＵＲＬへ行け」ボタン８２０（または、図１７に示すように、スーパーバイザ・セッション・インターフェースの「ＵＲＬへ行け」ボタン８６０）を選択する。

【０１５６】ステップ１９０６においては、（エージェント）マスター・アプレットまたは（スーパーバイザ）マスター・アプレットは、ＷＴＳサーバ１４４にコマンドを送る。

【０１５７】ステップ１９０８においては、ＷＴＳサーバ１４４は、マスター・アプレット１２４Ａおよび１２４Ｎに、ＵＲＬおよび時間情報と一緒にコマンドをシーケンシャルに送る。その結果、マスター・アプレット１２４Ａおよび１２４Ｎは、前記ＵＲＬに基づいて、ウェブ・ページをロードし、表示するように、その各ブラウザ１１４Ａおよび１１４Ｎに通知することができる。

【０１５８】ステップ１９１０においては、前記コマンドと一緒に送られる各ＵＲＬに対して、ＷＴＳサーバ１４４は、ブラウザ１１４Ａが、ＵＲＬ来歴リスト１およびデータ・リスト１に記憶した情報に基づいて、各ＵＲＬにより識別したウェブ・ページ上で、データ・フィールドに対して、ずでに何等かのアクティビティをしたかどうかをチェックする。

【０１５９】ステップ１９１２においては、ブラウザ１１４が、各ＵＲＬにより識別したウェブ・ページ上で、データ・フィールドに対して、すでに何等かのアクティビティをした場合には、ＷＴＳサーバ１４４は、（データ・フィールド名およびデータ・フィールドの数値と一緒に）（ブラウザ１１４Ａのところの）マスター・アプレット１２４Ａ、および（ブラウザ１１４Ｎのところの）マスター・アプレット１２４Ｎに、コマンドを送る。

【０１６０】ステップ１９１４においては、ブラウザ１１４Ａのところで、マスター・アプレット１２４Ａは、ＤＳＴアプレット１２６Ａに、コマンド、データ・フィールド名、およびデータ・フィールド数値を送る。その結果、ＤＴＳアプレット１２４Ａは、現在表示されているウェブ・ページ上の各データ・フィールドに、データ・フィールド値を表示することができる。ブラウザ１１４Ｎのところで、マスター・アプレット１２４Ｎは、ＤＳＴアプレット１２６Ｎに、コマンド、データ・フィールド名、およびデータ・フィールド数値を送る。その結果、ＤＴＳアプレット１２４Ｎは、現在表示されているウェブ・ページ上の各データ・フィールドに、データ・フィールド値を表示することができる。

【０１６１】ＵＲＬのロードおよびアンロード時間およびデータ・フィールドに対する設定時間は、ＵＲＬ来歴リスト１およびデータ・リスト１に記録されているので、そうしたい場合には、ＵＲＬにより識別されたすべてのウェブ・ページおよびデータ・フィールドに対して行われたアクティビティを時間情報に従って、コピーすることができる（ウェブ・ページのロード、ウェブ・ページ上でのデータ・フィールドの設定、およびウェブ・ページのアンロード）。

【０１６２】上記実施形態の場合には、ウェブ・ページに埋め込まれている、すべてのアプレット（マスター・アプレット、ＤＴＳアプレット、セッションＩＤアプレットおよびエージェント・アプレット）は、ジャバを使用して書き込まれる。しかし、前記アプレットのあるものまたはすべては、ジャバ・スクリプトのようなブラウザ・スクリプト語を使用して書き込むことができる。より詳細に説明すると、前記アプレットに対するコードは、ブラウザ・スクリプト語を使用して、ウェブ・ページに書き込むことができる。ウェブ・ブラウザは、ブラウザ・スクリプト語で書き込まれたアプレットを含む、ウェブ・ページをダウンロードする。ウェブ・ブラウザは、前記アプレットを前記ウェブ・ブラウザが動作している端末装置のメモリ領域に記憶し、その後それらを初期化し、呼び出す。

【０１６３】本発明は、下記の利点を持つ。

【０１６４】依存することができるウェブ・ページ追跡および同期　－　本発明は、複数のウェブ・ページがブラウザ・キャッシュまたはプロキシ・サーバ内に記憶されているキャシュされたページからのものであっても、すべてのユーザ・アクティビティを追跡し、同期する。

【０１６５】使用が簡単　：　本発明は、ウェブ・ナビゲーションを別々に再度生成している複数のユーザの現在の手動プロセスを必要としない。

【０１６６】実行が容易（ユーザの立場から見て）：本発明は、本発明をサポートするために、追加のソフトウェアを必要としない。ユーザは、ソフトウェアをインストールしたり、構成したり、または実行する必要がない。

【０１６７】携帯が可能　：　本発明は、クライアント側およびサーバ側のどちらでも、種々のオペレーティング・システム上で使用することができる。クライアント側では、ジャバ・アプレットをサポートするウェブ・ブラウザがありさえすればよい。サーバ側では、ＨＴＴＰサービスを供給する同じサーバ上にジャバ仮想機械（Java Virtual Machine、以下「ＪＶＭ」という）がありさえすればよい。実際には、各オペレーティング・システムに対するＪＶＭが存在するので、本発明のサーバ構成部分は、すべてのオペレーティング・システム上で動作できる可能性がある。

【0168】互換性　：　本発明は、種々のベンダーからの任意のHTTPサービスと一緒に動作する。何故なら、本発明のサーバ構成部分は、HTTPサービスによる処理を必要とせず、そのため、HTTPサービスから独立しているからである。

【0169】柔軟性　：　ウェブ・ページ同期を、ウェブ・ページ追跡と一緒に、独立して使用することができる。ウェブ・ページ同期は、追跡した任意のデータを永久に記憶する必要はない。

【0170】ユーザのプライバシー　：　本発明は、ユーザのプライバシーを妥当なレベルで確保する。何故なら、追跡および同期は、情報プロバイダが制御しているウェブサイトが提供するページだけに制限されるからである。

【0171】複数のHTTPサーバのサポート　：　本発明は、ある会社がそのウェブ・サイトで動作している複数の物理的サーバを持つ状態を処理することができる。何故なら、WTSゲートウェイおよびサーバ構成部分が独立しているので、ゲートウェイを各HTTPサーバ上に設置することができ、それぞれが共通のWTSサーバと通信することができるからである。

【0172】本発明を図面に詳細に示し、上記特許明細書内に詳細に説明してきたが、本発明は、その精神から逸脱することなしに他の実施形態により実行することができることを理解されたい。それ故、本発明の範囲は、本特許明細書内の図面および記述に制限されるものではなく、添付の特許請求の範囲により定義される。

【図面の簡単な説明】

【図1】本発明のN台の端末装置、ネットワーク、およびウェブ・サイトを含むシステムを示す図である。

【図2】本発明のN台の各端末装置が、その各マスター・アプレット、DTSアプレットおよびセッションIDアプレットをダウンロードした場合の状態を示す図である。

【図3】本発明の前記（消費者）マスター・アプレット、DTSアプレットおよびセッションIDアプレットをある端末装置にダウンロードするプロセスを示す図である。

【図4】前記（消費者）マスター・アプレット、DTSアプレットおよびセッションIDアプレットが、ある端末装置に予めダウンロードされ、キャッシュに記憶されたときに、本発明により動作をおこなうために、以降のウェブ・ページのロードに応じて前記アプレットが呼び出されるプロセスを示す図である。

【図5】前記（消費者）マスター・アプレット、DTSアプレットおよびセッションIDアプレットとウェブ・ページが、ある端末装置に予めダウンロードされ、キャッシュに記憶されたときに、本発明により動作を行うために、以降のウェブ・ページのロードに応じて前記アプレットが呼び出されるプロセスを示す図である。

【図6】本発明のより詳細なセッション表を示す図である。

【図7】本発明により、管理ページ貯蔵所149からエージェント・ページ（またはスーパバイザ・ページ）をダウンロードすることによって、あるエージェント（またはスーパーバイザ）が、セッション・インターフェースを生成する方法を示す図である。

【図8】本発明のあるエージェント・セッション・インターフェース、ブラウザ・スーパーバイザ・セッション・インターフェース及びスーパーバイザ・エージェント・セッション・インターフェースを示す図である。

【図9】本発明のあるエージェントが、あるセッションに参加する動作を示すフローチャートを示す図である。

【図10】本発明の二つのブラウザ例を示すスクリーン・ディスプレイを示す図である。

【図11】本発明のウェブ・ページ同期の動作を示すフローチャートを示す図である。

【図12】本発明の五つのデータ・フィールドを含むウェブ・ページ及び五つのデータ・フィールドの一つ内のデータが変化しているウェブ・ページを示す図である。

【図13】本発明のデータ同期の動作を示すフローチャートを示す図である。

【図14】本発明のウェブ・ページ追跡の動作を示すフローチャートを示す図である。

【図15】本発明のデータ追跡の動作を示すフローチャートを示す図である。

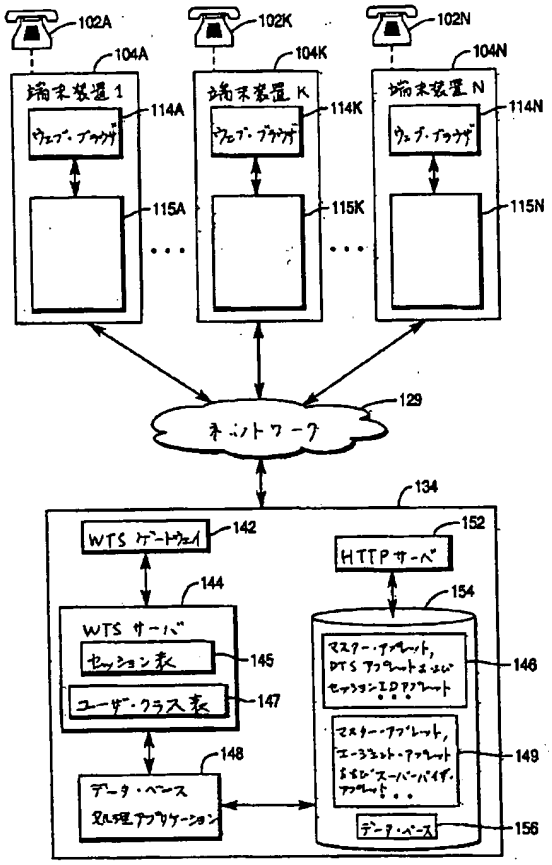
【図16】本発明のスーパーバイザのセッションへの参加の動作を示すフローチャートを示す図である。

【図17】本発明のあるスーパーバイザに対する三つのブラウザの例を示す図である。

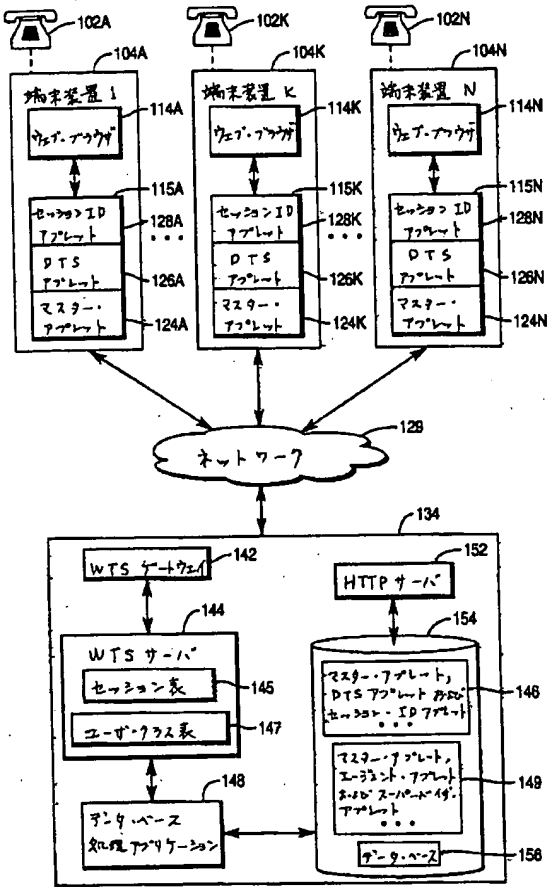
【図18】本発明のあるセッションにおいてすでに閲覧したウェブ・ページを再度ブラウズする動作を示すフローチャートを示す図である。

【図19】本発明のあるセッションにおいてすでに閲覧したすべてのウェブ・ページを再度ブラウズする動作を示すフローチャートを示す図である。

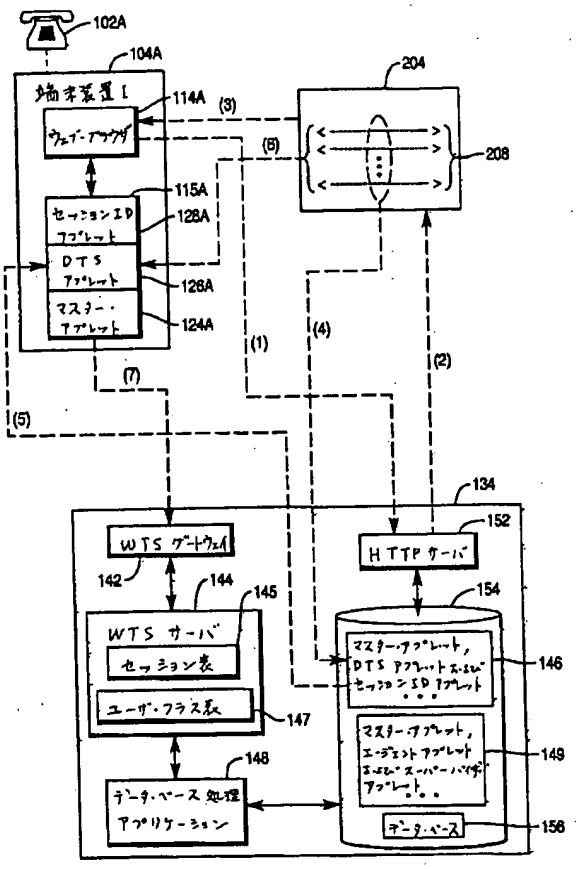
【図1】


102A
102K
102N
104A
104K
104N
端末装置1
端末装置K
端末装置N
ウェブ・ブラウザ
114A
114K
114N
115A
115K
115N
129
ネットワーク
134
WTS ゲートウェイ
142
HTTPサーバ
152
144
154
WTS サーバ
セッション表
145
ユーザ・クラス表
147
マスター・アプレット,
PTS アプレットおよび
セッションID アプレット
146
マスター・アプレット,
エージェント・アプレット
および スーパーバイザ・
アプレット・・・
149
データ・ベース
処理アプリケーション
148
データ・ベース
156

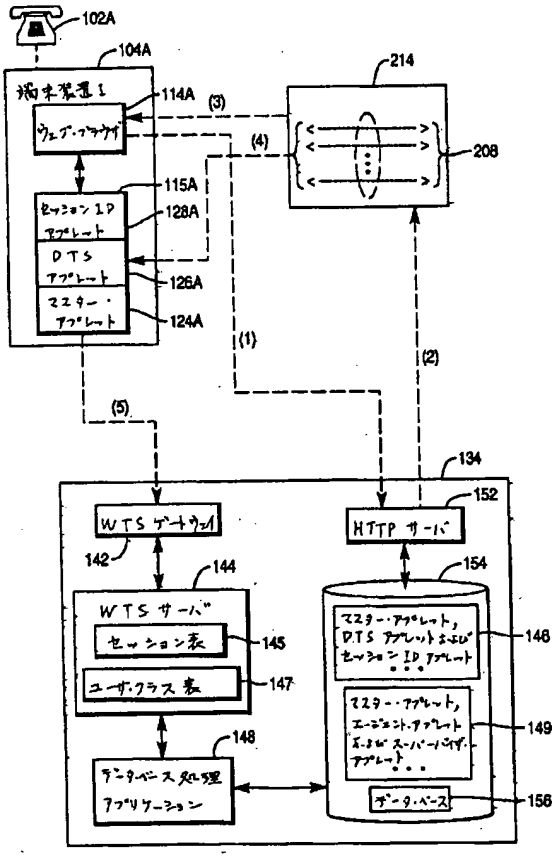
【図2】


102A
102K
102N
104A
104K
104N
端末装置 1
端末装置 K
端末装置 N
ウェブ・ブラウザ
114A
114K
114N
セッションID アプレット
DTS アプレット
マスター・アプレット
115A
128A
126A
124A
115K
128K
126K
124K
115N
128N
126N
124N
ネットワーク
129
134
WTS ゲートウェイ
142
HTTP サーバ
152
144
WTS サーバ
セッション表
145
ユーザ・クラス表
147
154
マスター・アプレット, DTS アプレット および セッション・ID アプレット
146
マスター・アプレット, エージェント・アプレット および スーパーバイザ・アプレット
149
148
データベース 処理アプリケーション
データベース
156

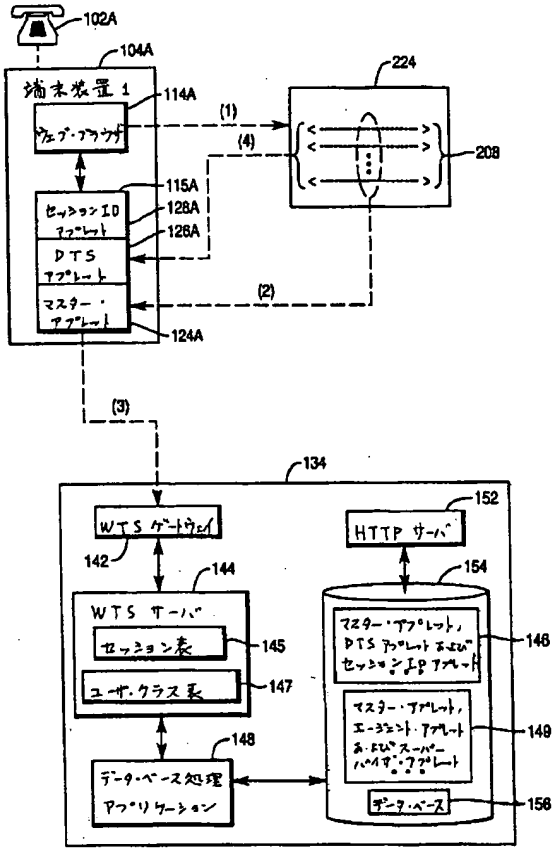
【図3】


102A
104A
端末装置1
114A
ウェブ・ブラウザ
(3)
(6)
204
208
115A
セッションID アプレット
128A
DTS アプレット
126A
マスター・アプレット
124A
(1)
(4)
(2)
(7)
(5)
134
WTS ゲートウェイ
142
144
145
WTS サーバ
セッション表
ユーザ・プロフィル表
147
148
データベース処理アプリケーション
HTTP サーバ
152
154
マスター・アプレット, DTS アプレットおよびセッションID アプレット
146
マスター・アプレット, エージェント・アプレットおよびスーパーバイザ・アプレット
149
データベース
156

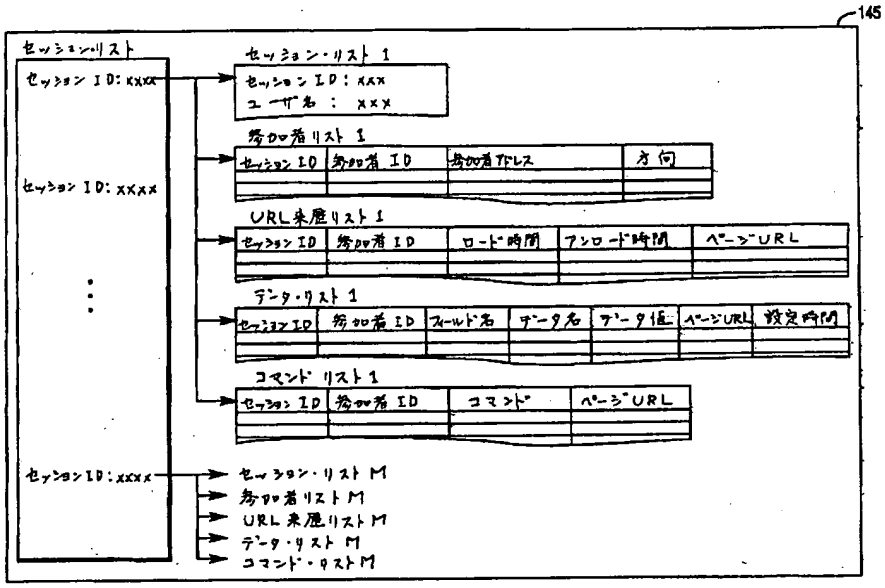
【図4】


102A
104A
端末装置1
114A
ウェブ・ブラウザ
(3)
(4)
214
208
115A
セッションID アプレット
128A
DTS アプレット
126A
マスター・アプレット
124A
(1)
(2)
(5)
134
WTS ゲートウェイ
142
HTTP サーバ
152
154
144
WTS サーバ
セッション表
145
ユーザクラス表
147
マスター・アプレット, DTS アプレットおよびセッションID アプレット
146
マスター・アプレット, エージェント・アプレット
149
データベース処理アプリケーション
148
データベース
156

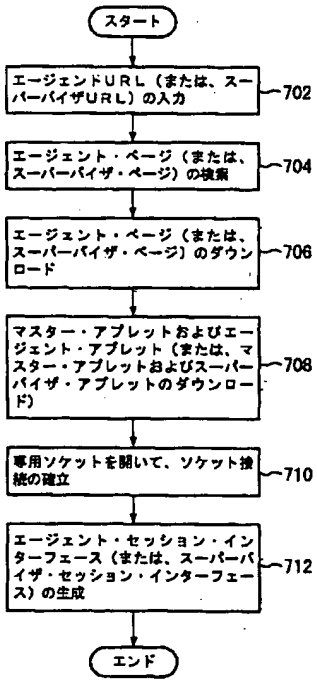
【図5】


102A
104A
端末装置 1
114A
ウェブ・ブラウザ
224
(1)
(4)
208
115A
セッションID アプレット
128A
DTS アプレット
126A
マスター・アプレット
124A
(2)
(3)
134
WTS ゲートウェイ
142
144
WTS サーバ
セッション表
145
ユーザ・クラス表
147
148
データベース処理 アプリケーション
152
HTTP サーバ
154
マスター・アプレット, DTS アプレットおよび セッションIP アプレット
146
マスター・アプレット, エージェント・アプレット および スーパーバイザ・アプレット
149
データベース
156

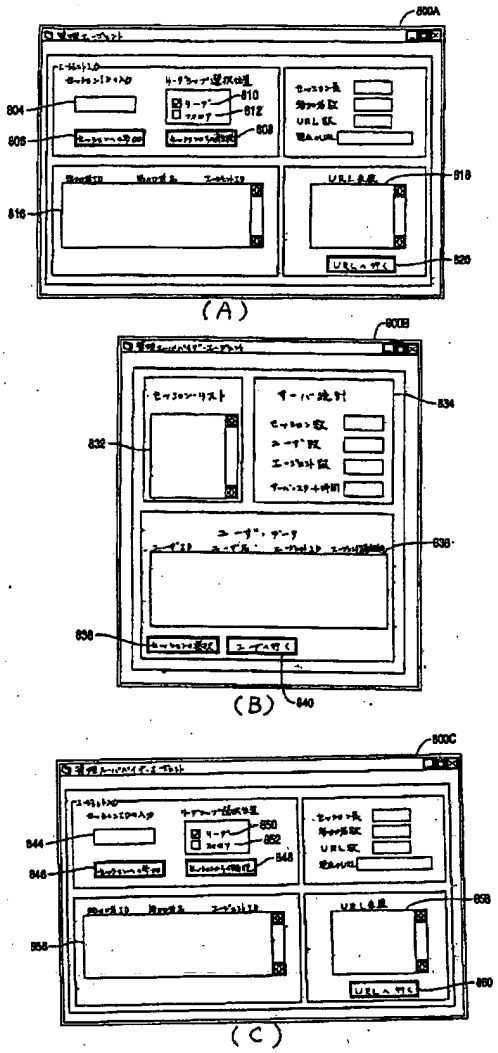
【図6】


145
セッションリスト
セッション ID: xxxx
セッション ID: xxxx
セッション ID: xxxx
セッション・リスト 1
セッション ID: xxx
ユーザ名 : xxx
参加者リスト 1
セッション ID
参加者 ID
参加者アドレス
方向
URL来歴リスト 1
セッション ID
参加者 ID
ロード時間
アンロード時間
ページURL
データ・リスト 1
セッション ID
参加者 ID
フィールド名
データ名
データ値
ページURL
設定時間
コマンド リスト 1
セッション ID
参加者 ID
コマンド
ページURL
セッション・リスト M
参加者リスト M
URL来歴リスト M
データ・リスト M
コマンド・リスト M

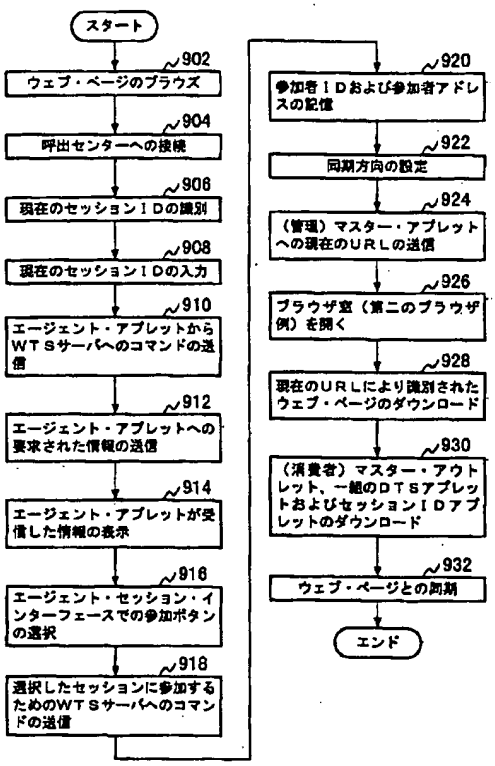
【図7】


スタート
エージェントURL（または、スーパーバイザURL）の入力
702
エージェント・ページ（または、スーパーバイザ・ページ）の検索
704
エージェント・ページ（または、スーパーバイザ・ページ）のダウンロード
706
マスター・アプレットおよびエージェント・アプレット（または、マスター・アプレットおよびスーパーバイザ・アプレットのダウンロード）
708
専用ソケットを開いて、ソケット接続の確立
710
エージェント・セッション・インターフェース（または、スーパーバイザ・セッション・インターフェース）の生成
712
エンド

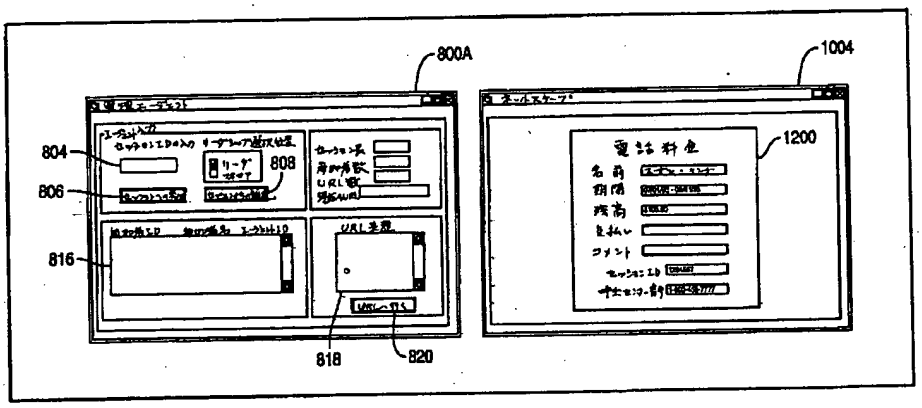
【図8】


800A
804
806
808
810
812
816
818
820
(A)
800B
832
834
836
838
840
(B)
800C
844
846
848
850
852
856
858
860
(C)

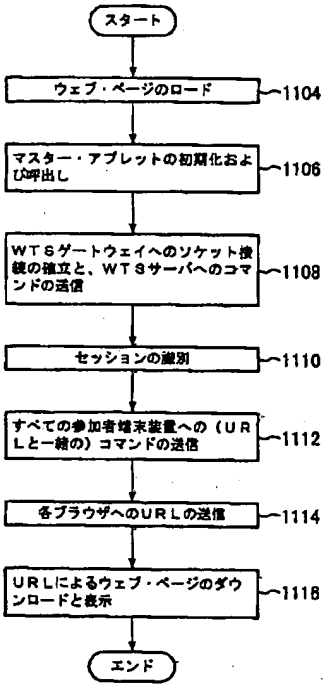
【図9】


スタート
902
ウェブ・ページのブラウズ
904
呼出センターへの接続
906
現在のセッションＩＤの識別
908
現在のセッションＩＤの入力
910
エージェント・アプレットからＷＴＳサーバへのコマンドの送信
912
エージェント・アプレットへの要求された情報の送信
914
エージェント・アプレットが受信した情報の表示
916
エージェント・セッション・インターフェースでの参加ボタンの選択
918
選択したセッションに参加するためのＷＴＳサーバへのコマンドの送信
920
参加者ＩＤおよび参加者アドレスの記憶
922
同期方向の設定
924
（管理）マスター・アプレットへの現在のＵＲＬの送信
926
ブラウザ窓（第二のブラウザ例）を開く
928
現在のＵＲＬにより識別されたウェブ・ページのダウンロード
930
（消費者）マスター・アウトレット、一組のＤＴＳアプレットおよびセッションＩＤアプレットのダウンロード
932
ウェブ・ページとの同期
エンド

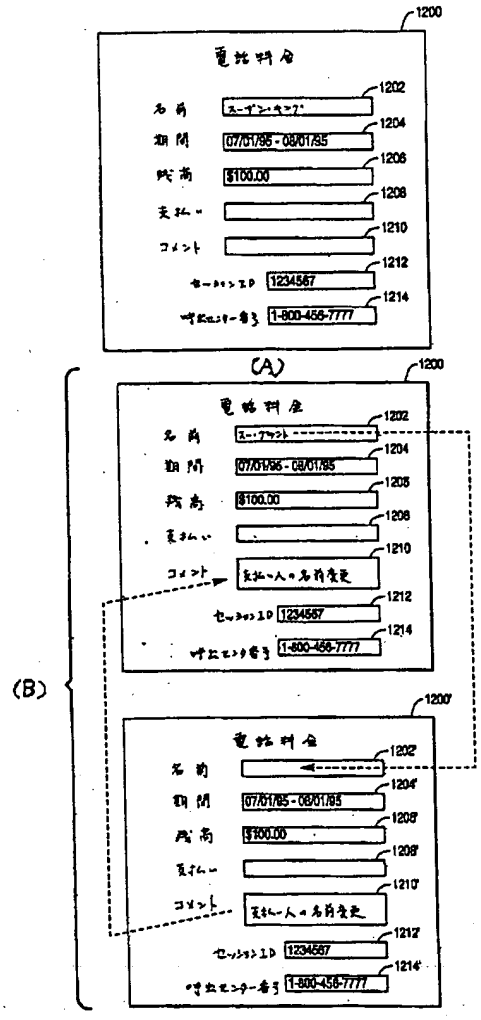
【図10】



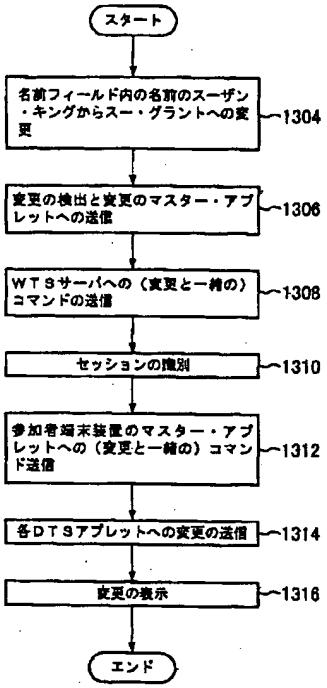
【図11】


スタート
ウェブ・ページのロード
1104
マスター・アプレットの初期化および呼出し
1106
WTSゲートウェイへのソケット接続の確立と、WTSサーバへのコマンドの送信
1108
セッションの識別
1110
すべての参加者端末装置への（URLと一緒の）コマンドの送信
1112
各ブラウザへのURLの送信
1114
URLによるウェブ・ページのダウンロードと表示
1116
エンド

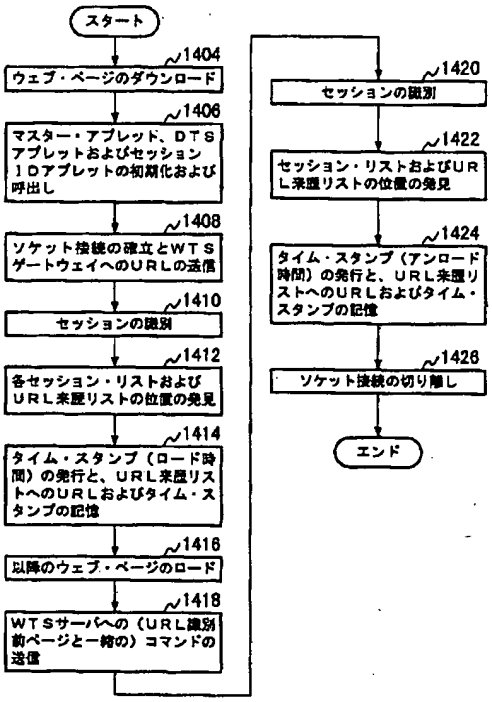
【図12】


1200
電話料金
名前
期間
残高
支払い
コメント
セッションID
呼出センター番号
1202
1204
1206
1208
1210
1212
1214
07/01/95 - 08/01/95
$100.00
1234567
1-800-456-7777
(A)
(B)
支払い人の名前変更

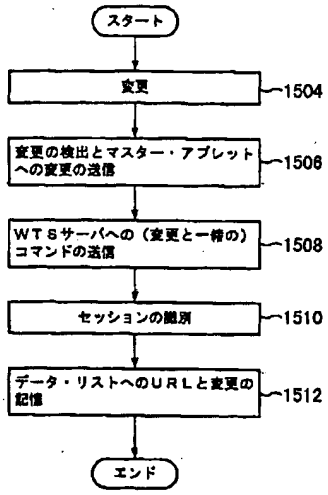
【図13】


スタート
名前フィールド内の名前のスーザン・キングからスー・グラントへの変更
1304
変更の検出と変更のマスター・アプレットへの送信
1306
WTSサーバへの（変更と一緒の）コマンドの送信
1308
セッションの識別
1310
参加者端末装置のマスター・アプレットへの（変更と一緒の）コマンド送信
1312
各DTSアプレットへの変更の送信
1314
変更の表示
1316
エンド

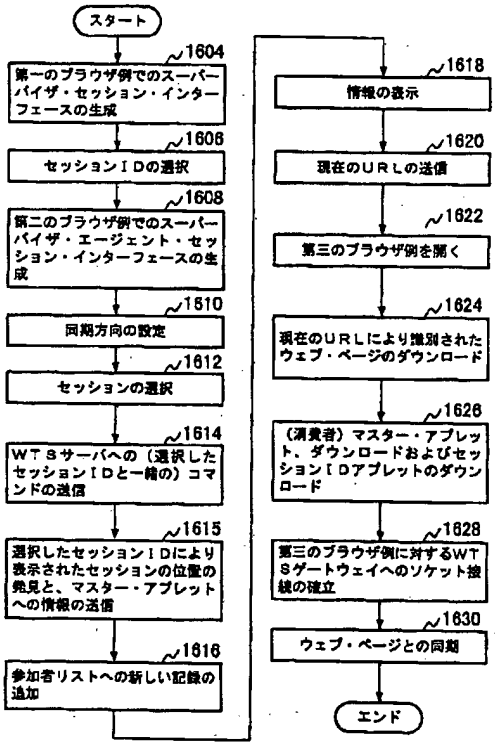
【図14】


スタート
1404
ウェブ・ページのダウンロード
1406
マスター・アプレット、DTSアプレットおよびセッションIDアプレットの初期化および呼出し
1408
ソケット接続の確立とWTSゲートウェイへのURLの送信
1410
セッションの識別
1412
各セッション・リストおよびURL来歴リストの位置の発見
1414
タイム・スタンプ（ロード時間）の発行と、URL来歴リストへのURLおよびタイム・スタンプの記憶
1416
以降のウェブ・ページのロード
1418
WTSサーバへの（URL識別前ページと一緒の）コマンドの送信
1420
セッションの識別
1422
セッション・リストおよびURL来歴リストの位置の発見
1424
タイム・スタンプ（アンロード時間）の発行と、URL来歴リストへのURLおよびタイム・スタンプの記憶
1426
ソケット接続の切り離し
エンド

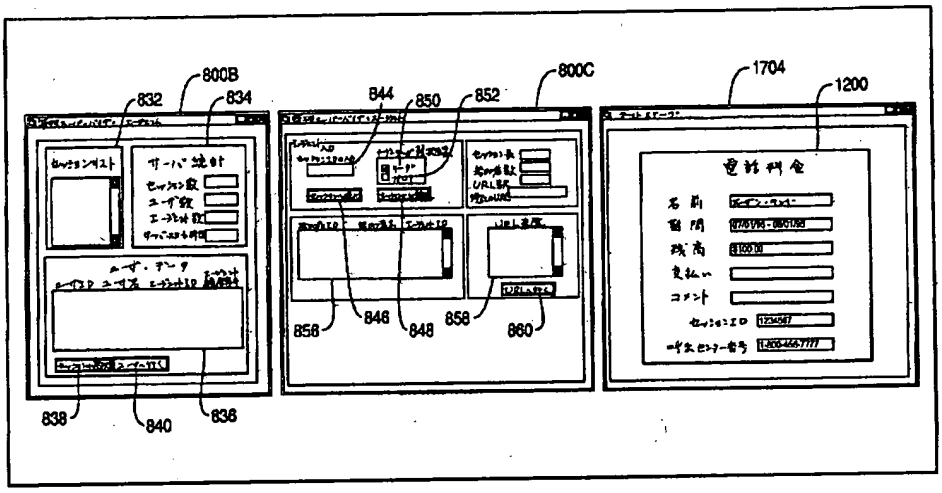
【図15】


スタート
変更
1504
変更の検出とマスター・アプレットへの変更の送信
1506
WTSサーバへの（変更と一緒の）コマンドの送信
1508
セッションの識別
1510
データ・リストへのURLと変更の記憶
1512
エンド

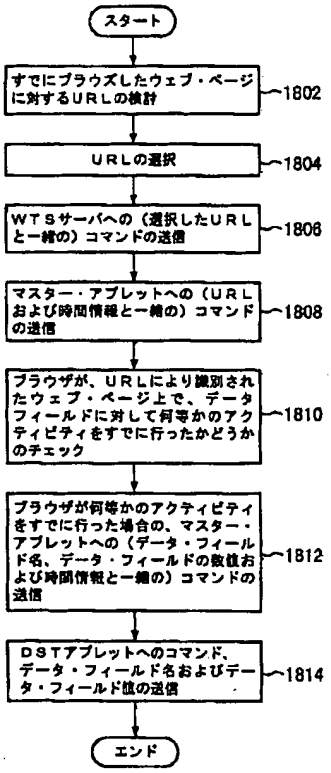
【図16】


スタート
1604
第一のブラウザ例でのスーパーバイザ・セッション・インターフェースの生成
1606
セッションＩＤの選択
1608
第二のブラウザ例でのスーパーバイザ・エージェント・セッション・インターフェースの生成
1610
同期方向の設定
1612
セッションの選択
1614
ＷＴＳサーバへの（選択したセッションＩＤと一緒の）コマンドの送信
1615
選択したセッションＩＤにより表示されたセッションの位置の発見と、マスター・アプレットへの情報の送信
1616
参加者リストへの新しい記録の追加
1618
情報の表示
1620
現在のＵＲＬの送信
1622
第三のブラウザ例を開く
1624
現在のＵＲＬにより識別されたウェブ・ページのダウンロード
1626
（消費者）マスター・アプレット、ダウンロードおよびセッションＩＤアプレットのダウンロード
1628
第三のブラウザ例に対するＷＴＳゲートウェイへのソケット接続の確立
1630
ウェブ・ページとの同期
エンド

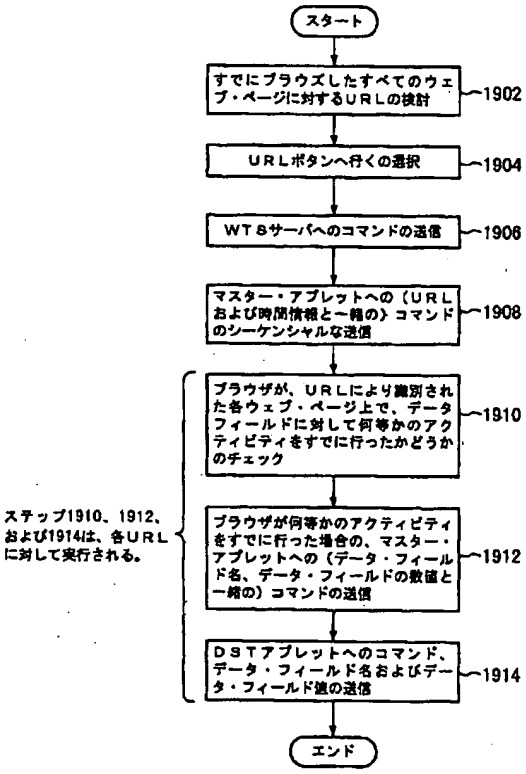
【図17】


800B
832
834
844
850
852
800C
1704
1200
838
840
836
856
846
848
858
860

【図18】


スタート
すでにブラウズしたウェブ・ページに対するURLの検討
1802
URLの選択
1804
WTSサーバへの（選択したURLと一緒の）コマンドの送信
1806
マスター・アプレットへの（URLおよび時間情報と一緒の）コマンドの送信
1808
ブラウザが、URLにより識別されたウェブ・ページ上で、データ・フィールドに対して何等かのアクティビティをすでに行ったかどうかのチェック
1810
ブラウザが何等かのアクティビティをすでに行った場合の、マスター・アプレットへの（データ・フィールド名、データ・フィールドの数値および時間情報と一緒の）コマンドの送信
1812
DSTアプレットへのコマンド、データ・フィールド名およびデータ・フィールド値の送信
1814
エンド

【図19】



フロントページの続き

(31)優先権主張番号　０８／９４４，７５７
(32)優先日　1997年10月６日
(33)優先権主張国　米国（ＵＳ）
(31)優先権主張番号　０８／９４４，７５９
(32)優先日　1997年10月６日
(33)優先権主張国　米国（ＵＳ）
(31)優先権主張番号　０８／９４４，９５１
(32)優先日　1997年10月６日
(33)優先権主張国　米国（ＵＳ）
(72)発明者　ミカエル　アイ　イングレイシア，Ｊｒ．
アメリカ合衆国　ニュージャージー州　08820フェザントラン　エディソン　63
(72)発明者　トーマス　エム　ローランド
アメリカ合衆国　ニュージャージー州　07704フェア　ヘブン　ケンブリッジ　アベニュー　232

【外国語明細書】

1 Title of Invention

MANAGING PAGE ACTIVITY OF WEB TERMINALS

2 Claims

1. A method for managing activities performed to pages [204,214,224...] at a terminal [104A] and activities between said terminal and a further terminal [104K, 104N], the pages being retrieved from a network site [134], characterised by the method comprising the steps of:

(a) at said terminal retrieving pages;

(b) at said terminal performing activities [114A, 115A] to said pages retrieved;

(c) at the network site recording [148] said activities for said terminal;

(d) at the further terminal establishing contact with said terminal; and

(e) at the further terminal displaying [Fig. 6] activities recorded for said terminal.

3 Detailed Description of Invention

The present invention relates generally to a method and apparatus for coordinating access to Internet web sites by a group of web browsers that are being run at a group of user terminals.

It is known that users can retrieve information from web sites (network sites) via the Internet. The basic model for retrieving information from web sites is user initiated information searching. Specifically, a user interacts with (via a terminal) a web browser to send a request to a web site. In response to the request, the web server for the web site retrieves the information requested and sends the web browser the information arranged in so called web page (HTML) format. One of the unique features of this model is the feature of "hyper-text links" embedded in web pages that have been retrieved. This feature enables a user in searching for information to "navigate" from one web page to another. In order to provide services (or assistance) to users (or customers) via the Internet, it is desirable to provide a mechanism to track activities performed to the web pages among a group of browsers.

One method of tracking web pages navigated is to install a monitoring program at a web site. When a terminal sends requests to a web site, the monitoring program at the web side collects the URLs(Uniform Resource Locator) for the requested web pages and sends the URLs to a server. However, under this method, the monitoring program is not always able to monitor the requests from the terminal, because when the terminal retrieves web pages from its browser cache space or from a proxy server, the requests are fulfilled locally and are never sent to the

web site. As a result, the URLs are not accurately tracked.

Another method of tracking web pages navigated is to install a monitoring program together with a browser at a terminal. The monitoring program constantly communicates with the web browser. However a monitoring program designed for a web browser manufactured by one vendor is typically not interoperable with or portable to another web browser manufactured by another vendor because browser interface mechanisms are proprietary. This results in a user needing a complicated monitoring program.

Therefore, there is a need for a technique to provide web page tracking and without requiring knowledge of the details about the web navigation software. Such a technique must operate where web page activity of itself does not generate a request to inform a web site of the activity.

According to the invention there is provided a method for managing activities performed to pages at a terminal and activities between said terminal and a further terminal, the pages being retrieved from a network site, comprising the steps of:

(a) at said terminal retrieving pages;

(b) at said terminal performing activities to said pages retrieved;

(c) at the network site recording said activities for said terminal;

(d) at the further terminal establishing contact with said terminal; and

(e) at the further terminal displaying activities recorded for said terminal.

The invention further provides the managing of page synchronization between activities performed either at said terminal or said further terminal. The activities may include loading a page, unloading a page and changing a data field on a page.

The further terminal may be an administrative terminal.

The invention provides a method for tracking activities with pages requested for loading over a network by retrieval from a page server at a network site to an individual terminal by sending information about the activities to a page tracking server comprising the steps of:

(g) loading a first page from the page server, the page being associated with a page locator for indicating a location of the page in the server, and the page containing program information of at least location information for indicating a program;

(h) loading the program from the server based on the location information, and executing the program;

(j) the program monitoring activities with the page; and

(k) the program sending information about the monitored activities to the page tracking server to be available for storage therein. The program information may be the actual program. The page locator may be URL (Uniform Resource Locator) and the program information may be a tag.

The invention will now be described by way of example with reference to the appended drawings, in which:

Detailed Description of the Preferred Embodiments

The following description is presented to enable any person skilled in the art to make and use the invention, and is provided in the context of a particular application and its requirements. Various modifications to the preferred embodiment(s) will be readily apparent to those skilled in the art, and the principles defined herein may be applied to other embodiments and applications without departing from the spirit and scope of the invention. Thus, the present invention is not intended to be limited to the embodiment(s) shown, but is to be accorded with the broadest scope consistent with the principles and features disclosed herein.

Referring to figure 1, there is shown an exemplary web page synchronization system 100, in accordance with the present invention.

As shown in figure 1, the system includes N terminals (104A, ▒, 104K, ▒, and 104N), a network 129 (the Internet, or a combination of the Internet and an Intranet), and a web

site 134. Each of the terminals has a telephone set (102A, ▒, 102K, ▒, or 102N) installed in its vicinity. Each of the terminals can be a PC computer, a workstation, a Java station, or even a web TV system.

Web site 134 includes a WTS (Web Tracking and Synching) gateway 142, a WTS server 144 containing a session table 145 and a user class table 147, a database processing application 148, an HTTP (Hyper Text Transfer Protocol) server 152, and a hard disk unit 154 for storing consumer page repository 146, administration page repository 149, and database 156. All the components in web site 134 can be installed in one or more computer systems. Each of the computer systems includes a processing unit (which may include a plurality of processors), a memory device, and a disk unit (which may include a plurality of disk sets).

Each of the terminals (104A, ▒, 104K, ▒, or 104N) includes a processor unit (not shown) and a memory area (115A, ▒, 115K, ▒, 115N), and runs a Java enabled web browser (114A, ▒, 114K, ▒, or 114N). Each of the memory area (115A, ▒, 115K, ▒, or 115N) is maintained by its respective browser (114A, ▒, 114K, ▒, or 114N). Via network 129, each of the browsers (114A, ▒, 114K, .., or 114N) is able to send requests to and receive web pages from HTTP server 152, and to display the web pages received at its respective terminal. Each of the browsers (114A, ▒, 114K, ▒, or 114N) is able to run a Master Applet (124A, ▒, 124K, ▒, or 124N), a set of DTS (Data Tracking and Synching) Applets, a SessionID Applet, and an Agent Applet. As shown

in figure 1, these Applets are stored in consumer page repository 146 and can be downloaded from consumer page repository 146 and stored in the memory areas of the terminals (104A, ▒, 104K, ▒, 104N).

Referring to figure 2, there is shown the situation where each of the terminals (104A, ▒, 104K, ▒, or 104N) has downloaded its respective Master Applets (124A, ▒, 124K, ▒, or 124N), DTS Applets (126A, ▒, 126K, ▒, or 128N), and SessionID Applet (128A, ▒, 128K, ▒, 128N), in accordance with the present invention.

In figure 2, each of the (consumer) Master Applet (124A, ▒, 124K, ▒, or 124N) is primarily responsible for: (1) in response to loading each web page at its respective browser, opening a dedicated socket, and establishing a socket connection to WTS gateway 142 via network 129 for its respective browser (114A, ▒, 114K, ▒, 114N), (2) communicating with WTS server 144 via the socket connection, from which WTS server 144 is able identify the origin (i.e. which browser, which web page, etc.) of the commands and information that are being delivered through, (3) monitoring the activities of its respective browser, (4) sending the information about its respective browser's activities to WTS server 144, (5) receiving and processing the information about other browsers' activities, (6) via the socket connection, providing a single communication path to WTS server 144 for DTS Applets (126A, ▒, 126K, ▒, or 126N), SessionID Applets (128A, ▒, 128K, ▒, or 128N), or any other consumer Applets embedded on the same page with the Master

Applet, (7) sending commands to WTS server 144 to request services, for itself and for DTS Applets (126A, ▒, 126K, ▒, or 126N), SessionID Applets (128A, ▒, 128K, ▒, or 128N) , or any other consumer Applets embedded on the same page with the Master Applet, and (8) sending user class information together with the commands, to indicate that its respective browser is a consumer user.

Each set of DTS Applets (126A, ▒, 126K, ▒, or 126N) contains one or more individual DTS Applets, which are primarily responsible for: (1) displaying and monitoring the data activities (data inputs or data updates of data fields) on web pages that are being displayed by its respective browser, (2) sending the data activities to WTS server 144 via its respective Master Applet, (3) receiving the data activities from other browsers via its respective Master Applet, and (4) processing the data activities from other browsers for the web pages that are being displayed by its respective browser.

Each of the SessionID Applets (128A, ▒, 128K, ▒, or 128N) is responsible for retrieving, and for displaying on a web page the current SessionID.

As shown in administration page repository 149, Agent Applet (or Supervisor Applet) is responsible for creating a session interface, joining, monitoring, and controlling a session through the session interface. The(administration) Master Applet is primarily responsible for: (1) opening a dedicated socket, and establishing a socket connection to WTS gateway 142 via network 129 for the session interface

created by Agent Applet, Supervisor Applet, or any other administration Applets embedded on the same web page with the Master Applet, (2) communicating with WTS server 144 via the socket connection, from which WTS server 144 is able identify the origin (i.e. from which session interface) of the commands and information that are being delivered through, (3) via the socket connection, providing a single communication path to WTS server 144 for Agent Applet, Supervisor Applet, or any other administration Applets embedded on the same web page with the Master Applet, and (4) sending user class information together with the commands, to indicate that its respective browser is an administration user.

WTS gateway 142 is responsible for maintaining all socket connections between Master Applets and WTS server 144. The connections between Master Applets and WTS gateway 142 take place using standard sockets. The connection between WTS gateway 142 and WTS server 144 takes place using RMI (Remote Method Invocation).

WTS server 144 is responsible for: (1) managing and tracking the activities of all browsers participating in active sessions, exemplary activities including: loading of, interacting with, and unloading of web pages, (2) recording the information about the activities, (3) managing the synchronization of the activities for all browsers participating in the active sessions, (4) creating a session when a consumer user (via a browser) sends a request to web site 134 for the first time, (5) defining the session length

intervals, (6) purging sessions that have been inactive for more than the specified session length intervals, (7) adding and deleting participants to a session, and (8) providing services to all commands from consumer Applets, such as: (consumer) Master Applet, DTS Applets, SessionID Applets, and administration Applets, such as: (administration) Master Applet, Agent Applets, and Supervisor Applet.

Consumer page repository 146 stores the web pages and Applets for consumers. Consumer Applets can be selectively embedded into consumer web pages. Exemplary consumer Applets include: (consumer) Master Applet, DTS Applets, SessionID Applet, etc.

Administration page repository 149 stores the web pages and Applets for call center administration users, including: administrator, supervisor, agent, etc. Administration Applets can be selectively embedded into administration web pages. Exemplary administration Applets include: (administration) Master Applet, Agent Applet, Supervisor Applet, etc.

To better describe the present invention, the Applets stored in (or downloaded from) repository 146 can be referred to as consumer Applets, and the Applets stored in (or downloaded from) repository 149 can be referred to as administration Applets. For example, the Master Applet stored in (or downloaded from) repository 146 can be referred to as consumer Master Applet, and the Master Applet stored in (or downloaded from) repository 149 can be referred to as administration Master Applets. HTTP server

152 contains a security application that allows consumer users to get access only to the web pages stored in consumer page repository 146, and allows administration users (such as administrator, supervisor, agent, etc.) to get access to the web pages stored in both consumer page repository 146 and administration page repository 149.

Session table 145 is responsible for maintaining the information for all active sessions.

Class table 147 is responsible for keeping records of user classes assigned to different users. Listed are exemplary user classes: administrator, supervisor, agent, and consumer.

Based on user classes (administrator, supervisor, agent, and consumer), WTS server 144 provides the following services:

(1) creating a session (consumer);

(2) storing data received from a session participant (supervisor, agent, and consumer);

(3) listing active sessions (administrator and supervisor);

(4) listing the information associated with active sessions (administrator, and supervisor);

(5) listing current users (administrator);

(6) joining a session (supervisor and agent);

(7) terminating a session (supervisor);

(8) monitoring a session (supervisor and agent);

(9) configuring a session parameters (administrator); and

(10) sending commands and information to a consumer Master Applet or an administration Master Applet in a participating browser (supervisor, agent, and consumer).

Database 156 is responsible for storing data collected in session table 145.

HTTP server 152 is responsible for processing the requests issued by one of the web browsers, retrieving the web pages from either consumer page repository 146 or administration page repository 149, and sending the web pages to the browsers that have generated these requests.

Database processing application 148 is responsible for writing the data collected in session table 145 into database 156.

Referring to figure 3, there is shown the process of how the (consumer) Master Applet, DTS Applets, and SessionID Applet being downloaded into terminal 104A from HTTP server 152 in response to loading an initial web page, and then being invoked to perform the operations in accordance with the present invention.

As shown in figure 3, a (consumer) Master Applet, a set of DTS Applets, and a SessionID Applet are embedded into web page 204 by using a set of applet tags 208. Web page 204 is associated with a specific URL indicating the location of web page 204 in HTTP server 152.

As indicated by dotted line (1), web browser 114A sends a request including the URL of web page 204 to HTTP server 152 via network 129. As indicated by dotted line (2), in response to the request, HTTP server 152 retrieves web page

204 from consumer page repository 146 and sends it to web browser 114A via network 129. Web page 204 contains a set of applet tags 208, which indicate the location of Master Applet, DTS Applets, and SessionID Applet in HTTP server 152. As indicated by dotted line (3), web browser 114A loads web page 204. As indicated by dotted line (4), since Master Applet, DTS Applets, and SessionID Applet have not been downloaded, web browser 114A sends requests via network 129, to download these Applets based on applet tags 208. As indicated by dotted line (5), HTTP server 152 sends Master Applet, DTS Applets, and SessionID Applet to browser 114A via network 129. As indicated by dotted line (6), browser 114A stores Master Applet 124A, DTS Applets 126A, and SessionID Applet 128A into memory area 115A of terminal 104A, and initializes and invokes these Applets. After being invoked, these Applets are running together with web browser 114A, to monitor and process the activities for which they are assigned to be responsible. As indicated by line (7), Master Applet 124A opens a dedicated socket and establishes a socket connection to WTS gateway 142 for browser 114A and web page 204. Via the socket connection, Master Applet 126 sends WTS server 144 a command, together with an ID unique to browser 114A. In response to the command from Master Applet 126, WTS server 144 creates a session for browser 114A based on the unique ID, and issues a time stamp (loading time) indicating the time at which the command was received, and stores the URL and time stamp of web page 204 into the session created for browser 114. As

will see in the description in connection with figure 6 following, the URL, command, and loading time are stored in a URL history list and a command list created for the session.

Referring to figure 4, there is shown the process of the (consumer) Master Applet 124A, DTS Applets 126A, and SessionID Applet 128A being invoked, in response to loading a subsequent web page 214 (subsequent to web page 204), to perform the operations in accordance with the present invention, when these Applets have been previously downloaded and cached in terminal 104A.

As indicated by dotted line (1), to download web page 214, web browser 114A sends a request including the URL of web page 214 to HTTP server 152 via network 129. Before loading web page 214, the following events occur: (a) browser 114A instructs Master Applet 124A to run a stop routine, (b) via the socket connection established for browser 114A and web page 204, Master Applet 124A sends a command to inform WTS server 144 that web page 204 has been unloaded, and disconnects the socket connection established for browser 114A and web page 204, (c) WTS server 144 issues a time stamp (unloading time) indicating the time the command was received, and (d) records the URL and the time stamp of web page 204 into the session created for browser 114A. As will be seen in the description in connection with figure 6, following, URL, command, and unloading time are stored in a URL history list and a command list created for the session. As indicated by dotted line (2), HTTP server

152 retrieves web page 214 from consumer page repository 146 and sends it to browser 114A. Like web page 204, web page 214 contains a set of applet tags 208 for indicating the location of Master Applet 124A, DTS Applets 126A, and SessionID Applet 128A. As indicated by dotted line (3), web browser 114A loads web page 214. As indicated by dotted line (4), in response to the loading of web page 214, web browser 114A locates Master Applet 124A, DTS Applets 126A, and SessionID Applet 128A (based on the indication of applet tags 208) that are cached by browser 114A in memory area 115A, and initializes these Applets and then invokes them. As indicated by line (5), Master Applet 124A opens a dedicated socket and establishes a socket connection to WTS gateway 142 for browser 114A and web page 214. Via the socket connection established for browser 114A and web page 214, Master Applet 126A sends a command, together with the ID unique to browser 114A and the URL of web page 214, to inform WTS server 144 that web page 214 has been loaded. WTS server 144 issues a time stamp (loading time) indicating the time the command was received and stores the URL and time stamp of web page into the session created for browser 114A. As will be seen in the description in connection with figure 6, following, URL, command, and loading time are stored in a URL history list and a command list created for the session.

Referring to figure 5, there is shown the process of the (consumer) Master Applet 124A, DTS Applets 126A, and SessionID Applet 128A being invoked, in response to loading

a subsequent web page 224 (subsequent to web page 214), to perform the operations in accordance with the present invention, when both these Applets and web page 224 have been previously downloaded and cached by browser 114A in terminal 104A.

As indicated by dotted line (1), web browser 114A loads web page 224 cached in memory area 115A maintained by browser 114A. Like web pages 204 and 214, web page 224 contains a set of applet tags 208 indicating the location of Master Applet 124A, DTS Applets 126A, and SessionID Applet 128A. Before loading web page 224, the following events occur: (a) browser 114A instructs Master Applet 124A to run a stop routine, (b) via the socket connection established for browser 114A and web page 214A, Master Applet 124A sends a command to inform WTS server 144 that web page 214 has been unloaded, and disconnects the socket connection established for browser 114A and web page 214, (c) WTS server 144 issues a time stamp (unloading time) indicating the time the command was received, and (d) WTS server 144 records the URL and time stamp of web page 214 into the session created for browser 114A. As will be seen in the description in connection with figure 6, following, the URL, command, and unloading time are stored in a URL history list and a command list created for the session. As indicated by dotted line (2), in response to the loading of web page 224, browser 114A locates Master Applet 124A, DTS Applets 126A, SessionID Applet 128A that have been cached by browser 114A in memory area 115A in terminal 104A, and initializes and

invokes these Applets. As indicated by line (3), Master Applet 124A opens a dedicated socket and establishes a socket connection to WTS gateway 142 for browser 114A and web page 224. Via the socket connection established for browser 114A and web page 224, Master Applet 126A sends a command, together with the ID unique to browser 114A and the URL of web page 224, to inform WTS server 144 that web page 224 has been loaded. WTS server 144 issues a time stamp (loading time) indicating the time the command was received and stores the URL and time stamp into the session created for browser 114. As will be seen in the description in connection with figure 6, following, the URL, command, and loading time are stored in a URL history list and a command list created for the session.

In the example shown in figure 5, it should be appreciated that even through no request arrives at HTTP server 144 when web page 224 is loaded from cached memory in terminal 104A, Master Applet 124A still sends browsing activities to WTS server 144.

It should be noted that the processes shown in figures 3-5 of loading and invoking Master Applet, DTS Applets, and SessionID Applet for terminal 104A can also be used for terminals 104K, ▒, 104N.

In figures 3-5, Master Applet, DTS Applets, and SessionID Applet are all embedded into web pages 204, 214, and 224. However, it should be noted that not all the Applets are required to be embedded into a web page. Depending on the desired functions to be performed,

respective Applets can be selectively embedded into a web page by selectively setting applet tags in the web page. For example, if data synchronization and tracking of individual elements are not needed, the applet tags for linking DTS Applets can be eliminated from the web page. By the same token, if additional functions are needed, additional applet tags can be added into the web page to link additional Applets.

Referring to figure 6, there is shown session table 145 (see figure 1) in greater detail, in accordance with the present invention.

While browsers at their respective terminals are browsing through the web pages in web site 134, WTS server 144 collects and analyzes the information about the interactions between all browsers and the web pages that have been downloaded to the browsers from web site 134. One difficulty in collecting and analyzing such information is that browsing individual web pages in web site 134 is a stateless process. More specifically, web site 134 receives a sequence of requests from different browsers, and sends the respective web pages to the respective browsers in response to the sequence requests. Since in processing the requests from an individual browser, web site 134 does maintain a constant connection to the same browser to keep an one-to-one relationship, web site 134 has no control over, or maintain data on, the sequences of the requests from the browsers.

To meaningfully collect and analyze the information about the interactions between the browsers and web pages, a session is defined as a collection of web page interactions that occur over a given period of time from a specific browser. A session is created when a browser first hits web site 134, and a session window (or session length interval) is defined for the session. If activities from a specific browser (identified by an ID unique to the browser, issued by a respective Master Applet) does not occur within the session window, the session is terminated and cleaned up by WTS server 144. A session window is refreshed (reset to time zero) each time the information about the associated browser is sent to WTS server 144. For example, if a session window is defined as 15 minutes, as long as the associated terminal has some activity every 15 minutes, the session will remain open. After 15 minutes of inactivity, the session is terminated and purged. A subsequent request from the same terminal will cause a new session to be created. After a session has been created for a terminal, one or more other terminals can join the session.

As shown in figure 6, session table 145 includes M Session IDs created for M sessions respectively. Each of the session ID is associated with: (1) a session list for maintaining information about a session, (2) a participant list for maintaining information about all participant browsers in a session (note: when a session is first created, it only contains one participant), (3) a URL history list for maintaining information about all web pages

visited by all participants in a session, (4) a data list for maintaining information about the data fields on the web pages visited by all participants in a session, and (5) a command list for maintaining information about all commands issued to WTS server 144 by the various participants in a session.

Typical items in a session list are: (1) SessionID for identifying a session, (2) UserName for indicating the actual name for whom the session is created, (3) StartTime for indicating the time of starting the session, (4) StopTime for indicating the time of stopping the session, and (5) SessionNotes for recording the notes of the session.

Typical fields contained in a participant list are: (1) SessionID for linking the participant list to a session, (2) ParticipantID for identifying a participant, (3) ParticipantAddresses for indicating a participant's IP address, (4) Class for indicating the user class of the participant (customer, agent, supervisor, administrator, etc.) and (5) Direction for indicating the synchronization direction for the participant browser.

Typical fields contained in a URL history list are: (1) SessionID for linking the URL history list to a session, (2) PageURL for indicating the URL of a web page visited, (3) ParticipantID for identifying a participant who visited the web page, (4) LoadingTime for indicating the loading time of the web page, and (4) UnloadingTime for indicating the unloading time of the web page.

Typical fields contained in a data list are: (1) SessionID for linking the data list to a session, (2) WasRelayed for indicating if this data field has been broadcasted, (2) FieldName for indicating the actual name of the data field, (3) DataName for indicating the name of the data field displayed on a web page, (4) DataValue for indicating the value of the data field, (5) TimeStamp for indicating the time at which this data field is updated, (6) URL for indicating the web page on which the data field was displayed, and (7) ParticipantID for indicating the participant browser who updated this data field.

Typical fields contained in a command list are: (1) SessionID for linking the data list to a session, (2) Command for indicating the specific command executed (loading a page, unloading a page, changing a data field, etc.), (3) URL for indicating the web page to which the command operated, (4) FieldPoint for indicating the data field to which the command operated, and (5) TimeStamp for indicating the time at which command was executed.

Before a session is purged from session table 145, database processing application 147 stores the associated session list, URL history list, and command list to database 156. The data contained in these three lists can be later used by data warehouse integration applications.

Referring to figure 7, there is shown an operation for creating a session interface for an agent (or a supervisor) by downloading an agent page (or a supervisor page) from administration page repository 149, in accordance with the

present invention. In the example shown in figure 7, it is assumed that administration user class (either agent user class or supervisor user class) is assigned to terminal 104N, so that the security application in HTTP server 152 grants the access to the web page stored in both consumer page repository 146 and administration page repository 149.

At step 702, an agent at terminal 104N types in an agent URL at terminal 104N, and browser 114N sends the URL to HTTP server 152, to retrieve an agent page, in which an (administration) Master Applet and an Agent Applet are embedded. For a supervisor, he/she types in a supervisor URL at terminal 104N, and browser 114N sends the URL to HTTP server 152, to retrieve a supervisor page, in which an (administration) Master Applet and a Supervisor Applet are embedded.

At step 704, HTTP server 152 retrieves the agent page (or a supervisor page) from administration page repository 149 and sends it to browser 114N.

At step 706, browser 114N downloads the agent page, in which a Master Applet (administration Master Applet) and an Agent Applet are embedded; or downloads the supervisor page, in which a Master Applet (administration Master Applet) and a Supervisor Applet are embedded.

At step 708, browser 114N downloads the Master Applet and Agent Applet from HTTP server 152, initializes and invokes these Applets; or downloads the Master Applet and Supervisor Applet from HTTP server 152, initializes and invokes these Applets.

At step 710, Master Applet opens a dedicated socket, establishes a socket connection to WTS gateway 142, and sends an ID unique to browser 114N to WTS server 144. WTS server 144 is able to identify browser 114N based on the unique ID.

At step 712, Agent Applet creates an agent session interface 800A shown in figure 8A for the agent user; or Supervisor Applet creates a supervisor session interface 800B shown in figure 8B for the supervisor agent.

Referring to figure 8A, there is shown an agent session interface 800A created for an agent at step 712, in accordance with the present invention.

As shown in figure 8A, the session interface contains a text box 804 for entering a session ID, a Join session button 806 for joining a session identified by the session ID, a drop button 808 for leaving a session, a leader check box 810 (selecting of which designates a browser as a leading browser in synchronization), a follower check box 812 (selecting of which designates a browser as a following browser in synchronization), a scrollable list box 816 for displaying the information contained in the participant list associated with a selected session, a scrollable list box 818 for displaying the information in an identified URL history list, and a text box 820 for displaying the information in an identified data list. If both the leader and follower check boxes 810 and 812 are selected in the agent session interface, browser 114A acts as both leading and following browser in synchronization.

Referring to figure 8B, there is shown a browser supervisor session interface 800B created for a supervisor at step 712, in accordance with the current invention.

As shown in figure 8B, the session interface contains a scrollable list box 832 for displaying session IDs of all active sessions in session table 145 and for selecting one of the session IDs, a text box 834 for displaying relevant statistics of WTS server 144, a multi column scrollable list box 836 for displaying details about the session selected in scrollable list box 832, a select session button 838 for selecting a session from scrollable list box 832. By using the information in scrollable list box 832, a supervisor agent can monitor all active sessions. By using the information in multi column scrollable list box 836, a supervisor can monitor operational status of a session selected from scrollable list box 832, including: (1) whether this session is being helped by an agent, (2) user name, and (3) agent ID. By selecting select session button 838, a supervisor can create an agent session interface as shown in figure 8C.

Referring to figure 8C, there is shown a supervisor agent session interface 800C, in accordance with the present invention.

Referring to figure 9, there is shown a flowchart illustrating the operation of joining a session by an agent, in accordance with the present invention.

In the example shown in figure 9, it is assumed that: (1) a consumer at terminal 104A is browsing web pages from

consumer page repository 146 via browser 114A, (2) session list 1 shown in figure 6 has been created for browser 114A, (3) an agent class has been assigned to browser 114N, (4) agent session interface 800A shown in figure 8A has been displayed on terminal 104N; (5) a (administration) Master Applet and Agent Applet have been previously downloaded into browser 114N, (6) a dedicated socket connection has been established for session interface 800A displayed at terminal 104N by the (administration) Master Applet, and (7) the agent at terminal 104A is on duty at a call center.

As shown in figure 9, at step 902, the consumer is browsing a web page at terminal 104A. On the web page, SessionID Applet 128A displays the current session ID. A call center telephone number the consumer can call is also displayed on the web page.

At step 904, the consumer is connected to the call center by dialing the telephone number via telephone 102A (see figure 1), and the call is directed by the call center to the agent.

At step 906, the consumer tells, via telephone 102A (see figure 1), the agent the current session ID displayed. It should be noted that, instead of using the telephone, the agent can be informed of the current session ID by alternative methods. For example, the consumer can enter his/her telephone number into a special web page that contains the caller ID of the consumer along with the current session ID. This information can be stored into a

special lookup table that can be used by the agent to lookup the current session ID.

At step 908, at terminal 104N, the agent types the current session ID into text box 804 (see figure 8A).

At step 910, in response to a loss of focus or a pressing of the Enter key, through the socket connection established for agent session interface 800A displayed on terminal 104N, the (administration) Master Applet at terminal 104N sends a command to WTS server 144, to retrieve the information in participant list 1, URL history list 1, and data list 1 (see figure 6) for the Agent Applet.

At step 912, WTS server 144 sends the information requested to the Agent Applet (via the Master Applet).

At step 914, the Agent Applet at terminal 104N displays some information from participant list 1 and URL history list 1 in (participant) scrollable list box 816 and (URL history) scrollable list box 818, respectively.

At step 916, the agent selects join button 806 in agent session interface 800A displayed on terminal 104N.

At step 918, in response to the selection at step 916, through the socket connection which has been established for agent session interface displayed on terminal 104N, the (administration) Master Applet sends WTS server 144 a command to join the selected session. Based on the identification associated with the socket connection, WTS server is able to generate a ParticipantID for browser 114N and to find the ParticipantAddress for terminal 104N.

At step 920, WTS server 144 stores the ParticipantID and ParticipantAddress into participant list 1. At this step, participant list 1 includes two participant records (two rows) containing the PaticipantIDs for browsers 114A and 114N respectively.

At step 922, at terminal 104N, the agent selects: leading check box 810 or following check box 812, or both of them. By only selecting leader check box 810, the activities at terminal 104N are synchronized at terminal 104A, but not other way around. By only selecting follower check box 812, the activities at terminal 104A are synchronized at terminal 104N, but not other way around. By selecting both leader and follower check boxes 810 and 812, the activities at terminals 104A and 104N are synchronized with each other (bi-directional synchronization). In response to the selection(s), through the socket connection which has been established for agent session interface 800A, the (administration) Master Applet sends WTS server 144 a command designating the synchronization direction. WTS server 144 stores the synchronization direction information into the Direction fields of the two records in participation list 1. In this example, it is assumed that the bi-directional synchronization has been selected for terminals 104A and 104N.

At step 924, WTS server 144 sends the (administration) Master Applet the URL of the web page being currently browsed at terminal 104A.

At step 926, the Agent Applet at terminal 104N opens a browser window 1004 (a second browser instance) as shown in figure 10.

At step 928, browser 114N downloads the web page identified by the URL from consumer page repository 146, and displays it in browser window 1004. A (consumer) Master Applet, a set of DTS Applets, and a SessionID Applet are embedded in the web page downloaded.

At step 930, browser 114N downloads (consumer) Master Applet 124N, set of DTS Applets 126N, and SessionID Applet 128N.

At step 932, the web pages displayed in second browser window 1004 at terminal 104N are being synchronized with the web pages being displayed at terminal 104A.

After step 932, if the agent (the first agent) at terminal 104N needs assistance from another agent (the second agent) at terminal 104K, the first agent can call the second agent and tell him/her the current session ID. The second agent can then join the current session using an agent session interface as shown in figure 8A displayed at terminal 104K.

Referring to figure 10, there is shown a screen display containing two browser instances (800A and 1004) at terminal 104N, in accordance with the present invention.

As shown in figure 10, at terminal 104N, the first browser instance provides an agent session interface 800A to control and monitor the current session, and the (administration) Master Applet for agent session interface

800A establishes and maintains a socket connection for agent session interface 800A. The second browser instance provides a browser window 1004 to display the web pages being synchronized. (Consumer) Master Applet 124N establishes and maintains a socket connection for each web page displayed in browser window 1004.

Referring to figure 11, there is shown a flowchart illustrating the operation of web page synchronization, in accordance with the present invention.

In the example shown in figure 11, it is assumed that: (1) a consumer at terminal 104A is browsing web pages from consumer page repository 146 via browser 114A, (2) a session has been created for browser 114A, (3) session list 1 and participant list 1 shown in figure 6 have been created for the session, (4) bi-directional synchronization has been selected for terminal 104A and all participant terminals, and (5) the (consumer) Master Applet, DTS Applets, and SessionID Applet have been downloaded into browser 104A and all participant browsers.

As shown in figure 11, at step 1104, browser 114A loads a web page either from consumer page repository 146 or from memory area 115A in terminal 104A. If Master Applet 124A, DTS Applets 126A, and SessionID Applet 128A had not been download to browser 114A, browser 114A would download these Applets from consumer page repository 146. However, in this example, these Applets are assumed to be downloaded.

At step 1106, in response to the loading of the web page, browser 114A initializes and invokes Master Applet 124A, DTS Applets 126A, and SessionID Applet 128A.

At step 1108, Master Applet 124A: (1) opens a dedicated socket, and establishes a socket connection to WTS gateway 142 for browser 114A and the web page loaded, and (2) via the socket connection, sends WTS server 144 a command together with an ID unique to browser 114A and the URL of the web page loaded. Based on the unique ID, WTS server is able to identify the session created for browser 114A.

At step 1110, WTS server 144 identifies the session for browser 114A.

At step 1112, WTS server 144 locates all IP addresses assigned to participant terminals in participant list 1 (shown in figure 6), and sends a command, together with the URL, to all the participant terminals (except that WTS server 144 does not sent the URL to terminal 104A, because the URL is originated from terminal 104A).

At step 1114, upon receiving the command, the (consumer) Master Applets in the participant terminals initialize themselves and pass the URL to their respective browsers.

At step 1116, the respective browsers in the participant terminals download and display the web page according to the URL.

It should be noted that, like terminal 104A, each of the participant terminals (at which agent session interface

is displayed) can lead the page synchronization using the operation shown in figure 11.

Referring to figure 12A, there is shown a web page containing five data fields, specifically: name 1202, time period 1204, account balance 1206, payment 1208, comments 1210, a text box 1212 for displaying the current session ID, and a text box for displaying the call center number the consumer can call, in accordance with the present invention.

Referring to figure 12B, there is shown a web page that is similar to that of figure 12A, except that the data in the field of name 202 is changed from Susan King to Sue Grant and the changes are synchronized at a participant terminal, in accordance with the present invention.

Referring to figure 13, there is shown a flowchart illustrating the operation of data synchronization, in accordance with the present invention.

In the example shown in figure 13, it is assumed that: (1) a customer at terminal 104A is browsing web pages via browser 114A, (2) a session has been created for terminal 104A, (3) session list 1 and participant list 1 shown in figure 6 has been created for the session, (4) terminal 104N is one of the participants, (5) web page 1200 containing five data fields shown in figure 12A is displayed on terminals 104A and all participant terminals, (6) a bi-directional synchronization has been selected for terminal 104A and all participant terminals, (7) the (consumer) Master Applet, DTS Applets, and SessionID Applet have been downloaded to browser 114A and the browsers at all

participant terminals, (8) the DTS Applets contains five individual Applets: DTS $Applet_1$, DTS $Applet_2$, DTS $Applet_3$, DTS $Applets_4$, and DTS $Applet_5$, (9) these five individual DTS Applets are respectively responsible for monitoring and processing the events occurred on the five data fields of web page 1200 shown in figure 12A, (10) (consumer) Master Applet 124A has established a dedicated socket connection to WTS gateway 142 for web page 12A displayed at terminal 104A, and (11) the customer at terminal 104A wants to make changes to name field 1202 from Susan King to Sue Grant.

As shown in figure 13, at step 1304, the customer changes the name in name field 1202 from Susan King to Sue Grant.

At step 1306, in response to a loss of focus on name field 1202 or pressing the Enter key, DTS $Applet_1$ detects the change and passes the change to Master Applet 124A.

At step 1308, via the dedicated socket connection, Master Applet 124A sends WTS server 144 a command together with the change of name field 1202. Since this change is passed to WTS server 144 via the dedicated socket connection established for web page 1200, WTS server 144 is able to recognize the origin of the command, web page 1200, and the name field upon which the change was made.

At step 1310, WTS server 144 identifies the session created for browser 114A.

At step 1312, WTS server 144 locates the IP addresses assigned to participant browsers in participant list 1 and sends a command (together with the change of name field

1202) to the Master Applets in all participant terminals (except that WTS server 144 does not send the command and change to browser 114A, since this change originated from browser 114A).

At step 1314, upon receiving the command, the (consumer) Master Applets (including Master Applets 124N) pass the change of name field 1200 to their respective DTS Applets, including the DTS $Applet_1$ at browser 114N.

At step 1316, the DTS $Applet_1$ display the update "Susan Grant" into the name fields on respective web page 1200 displayed on the respective terminals, including terminal 104N.

It should be noted that the operation shown in figure 13 can be used to perform data synchronization for the other four data fields on web page 1200 shown in figure 12A.

It should also be noted that the data field synchronization can also be performed at terminal 104N. For example, as shown in figure 12B, when the agent at terminal 104N enters comments of "Account's name had been changed" to comments field 1210' on web page 1200', this updates will be displayed in comments field 1210 at terminal 104A, by using the operation shown in figure 13.

Referring to figure 14, there is shown a flowchart illustrating the operation of web page tracking, in accordance with the present invention.

In the example shown in figure 14, it is assumed that: (1) a customer at terminal 104A is browsing web pages via browser 114A, (2) a session has been created for terminal

104A and all participant terminals, (3) session list 1, participant list 1, and URL history list 1 shown in figure 6 have been created for the session, (4) bi-directional synchronization has been selected for terminal 104A and all participant terminals, and (5) the (consumer) Master Applet, DTS Applets, and SessionID Applet have been downloaded into terminals 104A and all participant terminals.

As shown in figure 14, at step 1404, browser 114A downloads a web page from either the consumer page repository 146 or memory area 115A of terminal 104A. If Master Applet 124A, DTS Applets 126A, and SessionID Applet 128A had not been download to terminal 104A, browser 114A would download them from HTTP server 152. However, in this example, these Applets have been downloaded.

At step 1406, web browser 114A initializes and invokes Master Applet 124A, DTS Applets 126A, and SessionID Applet 128A.

At step 1408, Master Applet 124A opens a dedicated socket and establishes a socket connection to WTS gateway 142 for web browser 114A and the web page loaded. Master Applet 124A then sends WTS server 144 a command, together with: (1) an ID unique to browser 114A, and (2) the URL of the web page loaded. When commands and URL are delivered through this socket connection, WTS server 144 is able to recognize the origin of the commands and URL.

At step 1410, WTS server 144 identifies the session ID for browser 114A.

At step 1412, WTS server 144 locates the session list 1 and URL history list 1.

At step 1414, WTS server 144 issues a time stamp (loading time) for indicating the time at which the command was received, and stores the URL and time stamp to URL history list 1.

At step 1416, browser 114A sends WTS server 144 a request to load a subsequent web page.

At step 1418, before loading the subsequent web page, via the socket connection, Master Applet 124A sends WTS server 144 a command, together with the URL, to inform WTS server 144 that the current web page has been unloaded.

At step 1420, WTS server 144 identifies the session for terminal 104A.

At step 1422, WTS server 144 locates the session list 1 and URL history list 1.

At step 1424, WTS server 144 issues a time stamp (unloading time) for indicating the time at which the command was received, and stores the URL and time stamp to URL history list 1.

At step 1426, Master Applet 124A disconnects the socket connection for the web page that has been unloaded.

Referring to figure 15, there is shown a flowchart illustrating the operation of data tracking, in accordance with the present invention.

In the example shown in figure 15, it is assumed that: (1) a customer at terminal 104A is browsing web pages via browser 114A, (2) a session has been created for terminal

104A, (3) session list 1 and participant list 1 shown in figure 6 has been created for the session, (4) terminal 104N is one of the participants, (5) web page 1200 containing five data fields shown in figure 12A is displayed on terminals 104A and all participant terminals, (6) a bi-directional synchronization has been selected for terminal 104A and all participant terminals, (7) the (consumer) Master Applet, DTS Applets, and SessionID Applet are downloaded to terminal 104A and all participant terminals, (8) the DTS Applets contains five individual Applets: DTS $Applet_1$, DTS $Applet_2$, DTS $Applet_3$, DTS $Applets_4$, and DTS $Applet_5$, (9) these five individual DTS Applets are respectively responsible for displaying, monitoring and processing the events occurred on the five data fields of web page 1200 shown in figure 12A, (10) Master Applet 124A has established a dedicated socket connection to WTS server 144 for web page 1200 displayed on terminal 104A, and (11) the customer at terminal 104A wants to make changes to name field 1202 from Susan King to Sue Grant.

As shown in figure 15, at step 1504, the customer changes the name in name field 1502 from Susan King to Sue Grant.

At step 1506, in response to a loss of focus on name field 1202 or pressing the Enter key, DTS $Applet_1$ detects the change and passes the change to Master Applet 124A.

At step 1508, via the dedicated socket connection, Master Applet 124A sends WTS server 144 a command together with the change of name field 1202. Since this change is

passed to WTS server 144 via the dedicated socket connection established for web page 1200, WTS server 144 is able to recognize the origin of the command, web page 1200, and the name field upon which the change was made.

At step 1510, WTS server 144 identifies the session created for terminal 104A.

At step 1512, WTS server 144 stores the URL and update of name field 1202 into data list 1.

It should be noted that the operation shown in figure 15 can be used to perform data tracking for the other four data fields on web page 1200, and to perform data tracking for all participant terminals.

Referring to figure 16, there is shown a flowchart illustrating the operation of joining a session by a supervisor, in accordance with the present invention. In the example shown in figure 16, it is assumed that the supervisor is on duty at terminal 104K in a call center.

As shown in figure 16, at step 1604, the supervisor performs the steps shown in figure 7, where the supervisor downloads a supervisor page on which a (administration) Master Applet (referred as Master-Applet$_1$) and a Supervisor Applet are imbedded. The Supervisor Applet displays a supervisor session interface (as shown in figure 8B) in a first browser instance (see 800B in figure 17) on terminal 104K. Master-Applet$_1$ maintains a dedicated socket connection to WTS gateway 142 for the first browser instance (see 800B shown in figure 17).

At step 1606, from the first browser instance (see 800B shown in figure 17), the supervisor selects a session ID (listed in text box 832) and then select session button 838.

At step 1608, in response to the selection of the select session button 838, browser 114K downloads a supervisor agent page, on which a (administration) Master Applet (referred as Master-Applet$_2$) and an Agent Applet are embedded. The Agent Applet creates a supervisor agent session interface 800C (as shown in figure 8C) and displays it in a second browser instance (see 800C in figure 17) on terminal 104K. Master-Applet$_2$ opens and maintains a dedicated socket connection to WTS gateway 142 for the second browser instance (see supervisor agent session interface 800C shown in figure 17).

At step 1610, there can be two possible scenarios. A first scenario is that: an agent has joined the selected session to help the consumer, and the supervisor wants to join the selected session as a participant. Under this scenario, the supervisor simply selects join button 846, and leader and follower buttons 850 and 852 are both selected automatically. A second scenario is that: no agent has joined the session and the supervisor wants to join the session. Under this scenario, the supervisor joins the session just like an agent, by first selecting leader button 850 and/or follower button 852, and then join session button 846. In this example, it is assumed that a consumer at terminal 104A and an agent at terminal 104N have joined the

session, and the supervisor wants to join the selected session.

At step 1612, the supervisor selects join session button 846.

At step 1614, via the socket connection for the second browser instance (see supervisor agent session interface 800C shown in figure 17), the Master-Applet$_2$ sends WTS server 144 a command together with the selected session ID for the selected session.

At step 1615, WTS serer 144 locates the session indicated by the selected session ID, and sends information stored in participant list 1 and URL history list 1 (see figure 6) to Master-Applet$_2$.

At step 1616, WTS server 144 stores ParticipantID and ParticipantAddress into participant list 1 for browser 114K. At this step, participant list 1 includes three participant records (three rows) for browsers 114A, 114K, and 114N respectively.

At step 1618, Agent Applet at terminal 104K displays the information stored in participant list 1, and URL history list 1 in participant text box 856, and URL history text box 858 (see figure 8C) respectively.

At step 1620, WTS server 144 sends Master-Applet$_2$ the URL of the web page being currently displayed at terminals 104A and 104N.

At step 1622, the Agent Applet at terminal 104K opens a third browser instance (see 1704 in figure 17).

At step 1624, browser 114K downloads the web page identified by the URL from consumer page repository 146 (or loads the web page from memory area 115K in terminal 104K if it is cached there), and displays it in the third browser instance (see 1704 in figure 17).

At step 1626, browser 114K downloads (consumer) Master Applet 124K, DTS Applets 126K, and SessionID Applet 128 from consumer page repository 146 according to the applet tags in the current web page (assuming that these Applets have not previously downloaded).

At step 1628, Master Applet 124K opens a dedicated socket and establishes a socket connection to WTS gateway 142 for the third browser instance 1704 shown in figure 17.

After step 1630, the web pages displayed in third browser instance 1704 at terminal 104K are being synchronized with the web pages being displayed at terminals 104A and 104N.

Referring to figure 17, there is shown three browser instances (800B, 800C, and 1704) for the supervisor in response to the steps shown in figure 16, in accordance with the present invention.

Referring to figure 18, there is shown a flowchart illustrating the operation of re-browsing a web page previously reviewed in a session, in accordance with the present invention.

In the example shown in figure 18, it is assumed that: (1) a consumer at terminal 104A is browsing web pages from consumer page repository 146 via browser 114A, (2) a session

list 1 shown in figure 6 has been created for browser 114A, (3) an agent (or a supervisor) is on duty at terminal 104N in a call center, and agent class (or supervisor class) has been assigned to browser 104N, (4) at browser 114N, the first and second browser instances for the agent as shown in figure 10 (or the first, second and third browser instances for the supervisor as shown in figure 17) have been displayed, (5) via their respective socket connections established by their Master Applets, the first and second browser instances for the agent as shown in figure 10 (or the first, second and third browser instances for the supervisor as shown in figure 17) have been connected to WTS gateway 142, (6) Master Applets (124A and 124N), DTS Applets (126A and 126N) and SessionID Applets (128A and 128N) have been downloaded into terminals 104A and 104N respectively, (7) the agent (or supervisor) has selected and joined the session created for browser 114A, (8) at browser 114N, the second browser instance for the agent as shown in figure 10 (or the third browser instance for the supervisor as shown in figure 17) is being synchronized with browser 114A, and (9) bi-direction synchronization has been selected for browsers 114A and 114N.

At step 1802, for an agent user, via scrollable list box 818 on agent session interface shown in figure 10, he/she reviews the URLs for all the web pages previously browsed by browser 114A in the selected session. For a supervisor, via scrollable list box 858 on supervisor session interface shown in figure 17, he/she reviews the

URLs for all the web pages previously browsed by browser 114A in the selected session.

At step 1804, to display an individual web page previously browsed by browser 114A, the agent (or supervisor) selects a URL from scrollable list box 818 (or scrollable list box 858) and double-clicks on it.

At step 1806, for the agent, the (agent) Master Applet or (the supervisor Master Applet) sends WTS server 144 a command together with the selected URL, via its respective socket connection.

At 1808, WTS server 144 sends a command together with the URL and the time information (loading and unloading) to Master Applets 124A and 124N, so that Master Applets 124A and 124N can inform their respective browsers 114A and 114N to load and display the web page based on the URL.

At 1810, WTS server 144 checks whether browser 114A previously performed any activities to data fields on the web page identified by the URL, based on the information stored in URL history list 1 and data list 1. As shown in figure 6, URL history 1 contains the information about: (a) participant ID of browser 114A, (b) the URL, and (c) the loading and unloading time of the web page identified by the URL. Data list 1 contains the information about: (a) data field names for data fields, (b) value of the data fields, and (c) the times at which values of the data fields were changed.

At step 1812, if browser 114 previously performed any activities to the data fields on the web page identified by

the URL, WTS server 144 sends a command (together with the data field names, values of the data fields, and time information) to Master Applet 124A (at browser 114A) and Master Applet 124N (at browser 114N).

At step 1814, at browser 114A, Master Applet 124A passes the command, data field names, and data field values to DST Applets 126A, so that DTS Applets 124A can display the data field values into respective data fields on the web page being displayed. At browser 114N, Master Applet 124N passes the command, data field names, and data field values to DST Applets 126N, so that DTS Applets 124N can display the data field values into respective data fields on the web page being displayed.

Since the loading time and unloading time of the URL and the setting time for a data field are recorded in URL history list 1 and data list 1, if desired, the web page identified by the URL and the activities performed to the data fields can be duplicated (loading the web page, setting data fields on the web page, and unloading the web page) according to the time information.

Referring to figure 19, there is shown a flowchart illustrating the operation of re-browsing all web pages previously reviewed in a session, in accordance with the present invention.

In the example shown in figure 19, it is assumed that: (1) a consumer at terminal 104A is browsing web pages from consumer page repository 146 via browser 114A, (2) a session list 1 shown in figure 6 has been created for browser 114A,

(3) an agent (or a supervisor) is on duty at terminal 104N in a call center, and agent class (or supervisor class) has been assigned to browser 104N, (4) at browser 114N, the first and second browser instances for the agent as shown in figure 10 (or the first, second and third browser instances for the supervisor as shown in figure 17) have been displayed, (5) via their respective socket connections established by their respective Master Applets, the first and second browser instances for the agent as shown in figure 10 (or the first, second and third browser instances for the supervisor as shown in figure 17) have been connected to WTS gateway 142, (6) Master Applets (124A and 124N), DTS Applets (126A and 126N) and SessionID Applets (128A and 128N) have been downloaded into terminals 104A and 104N respectively, (7) the agent (or supervisor) has selected and joined the session created for browser 114A, (8) at browser 114N, the second browser instance for the agent as shown in figure 10 (or the third browser instance for the supervisor as shown in figure 17) is being synchronized with browser 114A, and (9) bi-direction synchronization has been selected for browsers 114A and 114N.

At step 1902, for an agent user, via scrollable list box 818 on agent session interface shown in figure 10, he/she reviews the URLs for all the web pages previously browsed by browser 114A in the selected session. For a supervisor, via scrollable list box 858 on supervisor session interface shown in figure 17, he/she reviews the

URLs for all the web pages previously browsed by browser 114A in the selected session.

At step 1904, to display all web pages previously browsed by browser 114A, the agent (or supervisor) selects Go to URLs button 820 in the agent session interface as shown in figure 10 (or Go to URLs button 860 in the supervisor session interface as shown in figure 17.

At step 1906, the (agent) Master Applet, or the (supervisor) Master Applet, sends a command to WTS server 144.

At 1908, WTS server 144 sequentially sends commands, together the URLs and time information, to Master Applets 124A and 124N, so that Master Applets 124A and 124N can inform their respective browsers 114A and 114N to load and display the web pages based on the URLs.

At 1910, for each one of URLs that are sent together with the commands, WTS server 144 checks whether browser 114A previously performed any activities to data fields on a web page identified by a respective URL, based on the information stored in URL history list 1 and data list 1.

At step 1912, if browser 114 previously performed any activities to the data fields on the web page identified by a respective URL, WTS server 144 sends a command (together with the data field names and values of the data fields) to Master Applets 124A (at browser 114A) and Master Applet 124N (at browser 114N).

At step 1914, at browser 114A, Master Applet 124A passes the command, data field names, and data field values

to DST Applets 126A, so that DTS Applets 124A can display the data field values into respective data fields on the web page being currently displayed. At browser 114N, Master Applet 124N passes the command, data field names, and data field values to DST Applets 126N, so that DTS Applets 124N can display the data field values into respective data fields on the web page being currently displayed.

Since the loading time and unloading time of the URLs and the setting time for data fields are recorded in URL history list 1 and data list 1, if desired, all the web pages identified by the URLs and the activities performed to the data fields can be duplicated (loading the web page, setting data fields on the web page, and unloading the web page) according to the timing information.

It should be noted that, in the above-described embodiments, all the Applets (Master Applets, DTS Applets, SessionID Applets, and Agent Applet) embedded into web pages are written using Java. However, some or all of these Applets can be written using a browser script language, such as Java Script. More specifically, the codes for these Applets can be selectively written into web pages using the browser script language, instead of using applet tags to link these Applets. When a web browser downloads a web page containing the Applets written in browser script language, it stores these Applets into the memory area of the terminal on which the web browser is running, and then initializes and invokes them.

The present invention has the following advantages:

Dependable web page tracking and synchronizing - It tracks and synchronizes all user activities, even if web pages come from cached pages stored in browser cache or proxy servers.

Ease of use - It eliminates the current manual process of multiple users separately re-creating the web navigation.

Ease of execution (from users' point of view) - It does not require additional software to support the present invention. No software needs to be installed, configured, or run by a user.

Portability - It works across different operating systems at both client and server sides. On the client side, the requirement is that there be a web browser that supports Java Applets. On the server side, the requirement is that there be a Java Virtual Machine (JVM) on the same server that provides the HTTP service. Since there are JVMs practically for every operating system, the server components of the present invention have the potential to run on all the operating systems.

Compatibility - It works together with any HTTP servers from different vendors because the server components of the present invention requires no processing by HTTP servers, and thus are independent from HTTP servers.

Flexibility - Web page synchronization can be used independently in conjunction with web page tracking. Web

page synchronization does not require persistent storage of any of the data tracked.

User privacy - It ensures a reasonable level of user privacy, since tracking and synchronization is limited to pages served by a web site that the information provider has control over.

Multiple HTTP server supported - It can handle the situation where a company has multiple physical servers running its web site, since the separation of the WTS gateway and server components enables a gateway to be placed on each HTTP server -- each communicating with a common WTS server.

While the invention has been illustrated and described in detail in the drawing and foregoing description, it should be understood that the invention may be implemented through alternative embodiments within the spirit of the present invention. Thus, the scope of the invention is not intended to be limited to the illustration and description in this specification, but is to be defined by the appended claims.

4 Brief Description of Drawings

figure 1 shows a system includes N terminals, a network, and a web site, in accordance with the present invention;

figure 2 shows a situation where each of the N terminals has downloaded its respective Master Applets, DTS Applets, and SessionID Applet, in accordance with the present invention;

figure 3 shows the process the (consumer) Master Applet, DTS Applets, and SessionID Applet being downloaded into a terminal, in accordance with the present invention;

figure 4 shows the process the (consumer) Master Applet, DTS Applets, and SessionID Applet being invoked, in response to loading a subsequent web page, to perform the operations in accordance with the present invention, when these Applets have been previously downloaded and cached in a terminal;

figure 5 shows the process of the (consumer) Master Applet, DTS Applets, and SessionID Applet being invoked, in response to loading a subsequent web page, to perform the operations in accordance with the present invention, when both these Applets and the web page have been previously downloaded and cached in a terminal;

figure 6 shows a session table in greater detail, in accordance with the present invention;

figure 7 shows how an agent (or supervisor) can create a session interface by downloading an agent page (or a supervisor page) from administration page repository 149, in accordance with the present invention.

figure 8A shows an agent session interface, in accordance with the present invention;

figure 8B shows a browser supervisor session interface, in accordance with the current invention;

figure 8C shows a supervisor agent session interface, in accordance with the present invention;

figure 9 shows a flowchart illustrating the operation of joining a session by an agent, in accordance with the present invention;

figure 10 shows a screen display containing two browse instances, in accordance with the present invention;

figure 11 shows a flowchart illustrating the operation of web page synchronization, in accordance with the present invention;

figure 12A shows a web page containing five data fields, in accordance with the present invention.

figure 12B shows a web page that is similar to that of figure 12A, except that the data in one of the five data fields is changed, in accordance with the present invention;

figure 13 shows a flowchart illustrating the operation of data synchronization, in accordance with the present invention;

figure 14 shows a flowchart illustrating the operation of web page tracking, in accordance with the present invention;

figure 15 shows a flowchart illustrating the operation of data tracking, in accordance with the present invention;

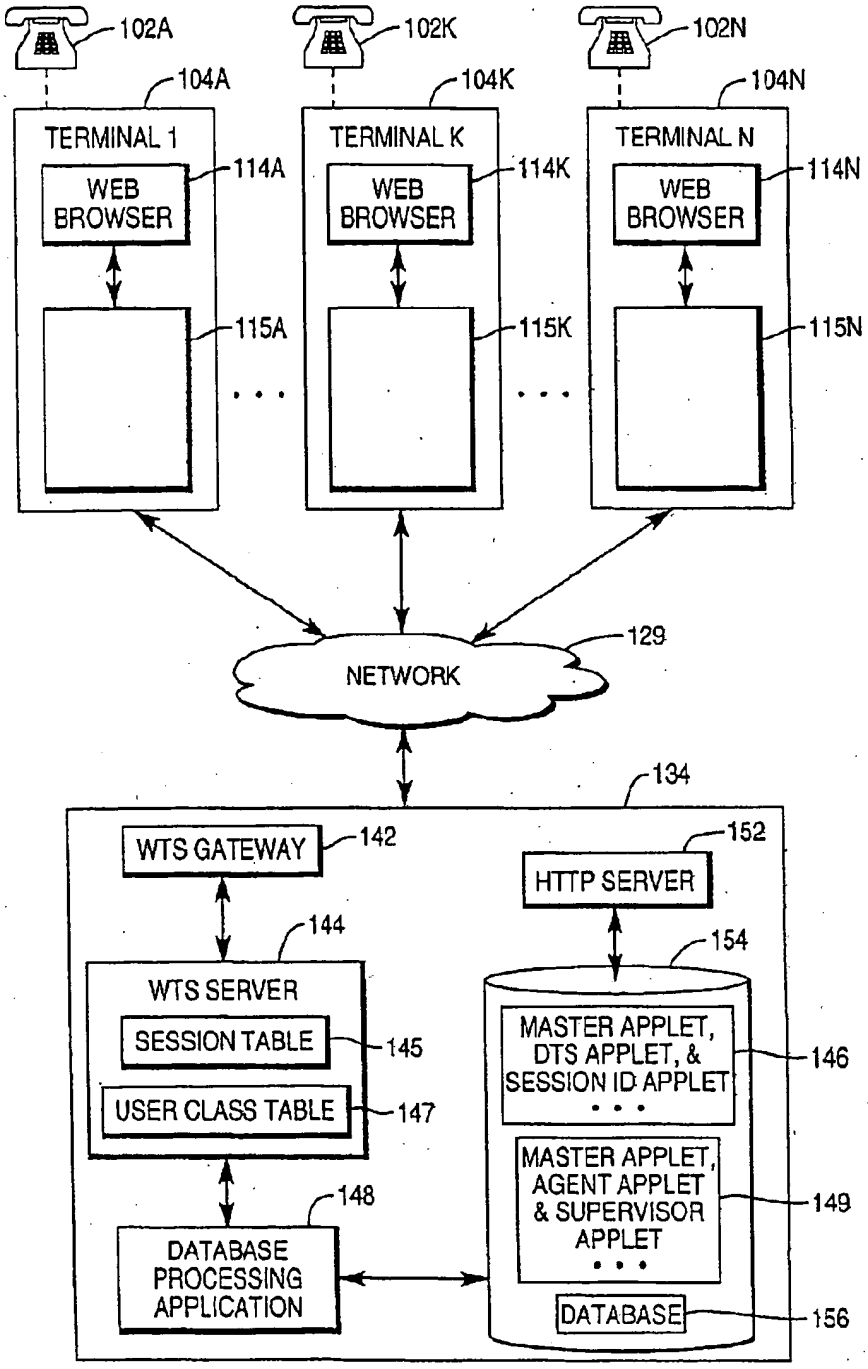
figure 16 shows a flowchart illustrating the operation of joining a session by a supervisor, in accordance with the present invention.

figure 17, there shows three browser instances for a supervisor, in accordance with the present invention;

figure 18 shows a flowchart illustrating the operation of re-browsing a web page previously viewed in a session, in accordance with the present invention; and

figure 19 shows a flowchart illustrating the operation of re-browsing all web pages previously viewed in a session, in accordance with the present invention.

FIG. 1 1/22


102A
102K
102N
104A
104K
104N
TERMINAL 1
TERMINAL K
TERMINAL N
WEB BROWSER
114A
114K
114N
115A
115K
115N
NETWORK
129
134
WTS GATEWAY
142
HTTP SERVER
152
144
154
WTS SERVER
SESSION TABLE
145
USER CLASS TABLE
147
MASTER APPLET, DTS APPLET, & SESSION ID APPLET
146
MASTER APPLET, AGENT APPLET & SUPERVISOR APPLET
149
148
DATABASE PROCESSING APPLICATION
DATABASE
156

2/22

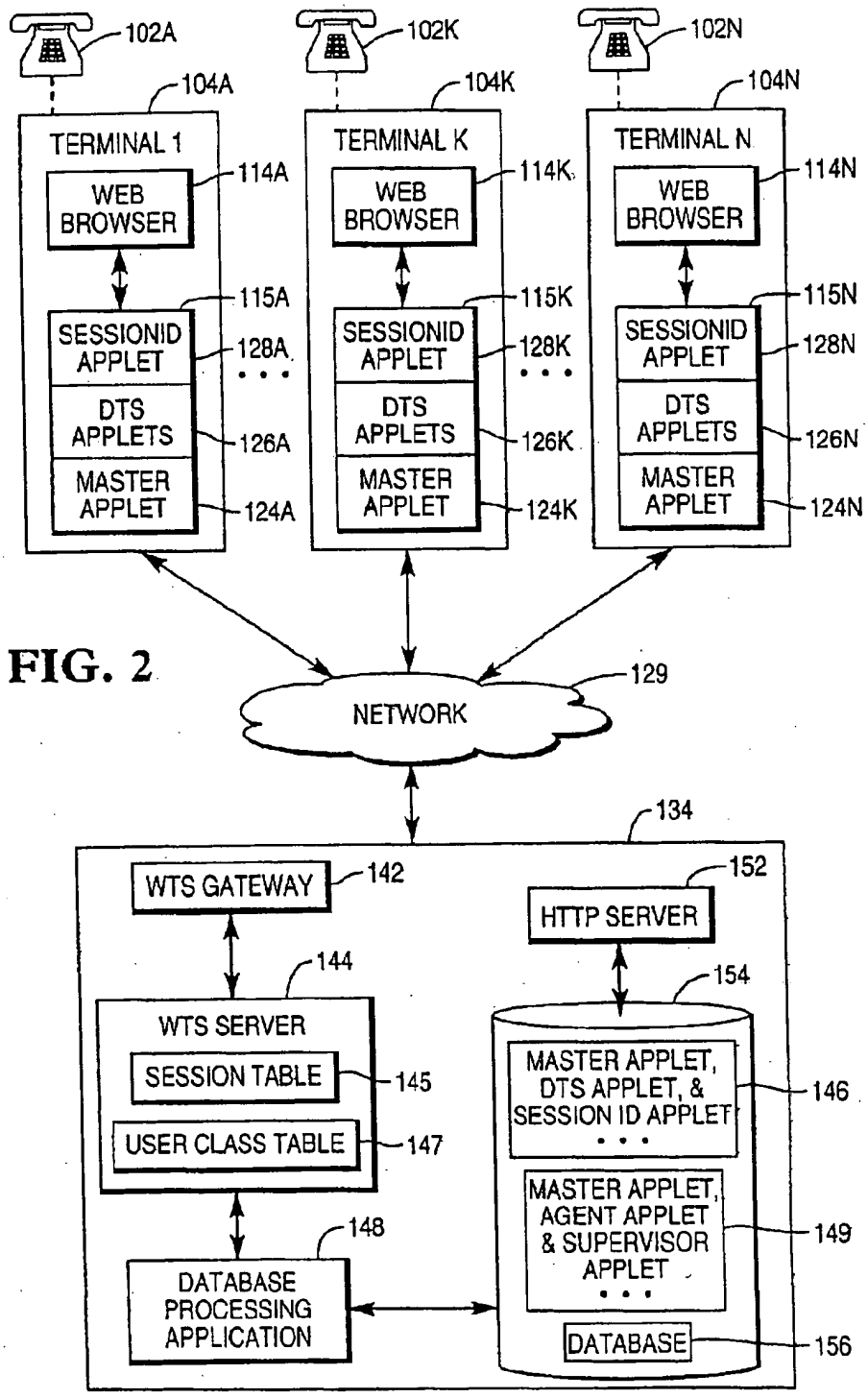

102A
102K
102N
104A
104K
104N
TERMINAL 1
TERMINAL K
TERMINAL N
WEB BROWSER
114A
114K
114N
SESSIONID APPLET
115A
115K
115N
128A
128K
128N
DTS APPLETS
126A
126K
126N
MASTER APPLET
124A
124K
124N
NETWORK
129
134
WTS GATEWAY
142
HTTP SERVER
152
144
154
WTS SERVER
SESSION TABLE
145
USER CLASS TABLE
147
MASTER APPLET, DTS APPLET, & SESSION ID APPLET
146
MASTER APPLET, AGENT APPLET & SUPERVISOR APPLET
149
148
DATABASE PROCESSING APPLICATION
DATABASE
156

FIG. 2

3/22

FIG. 3

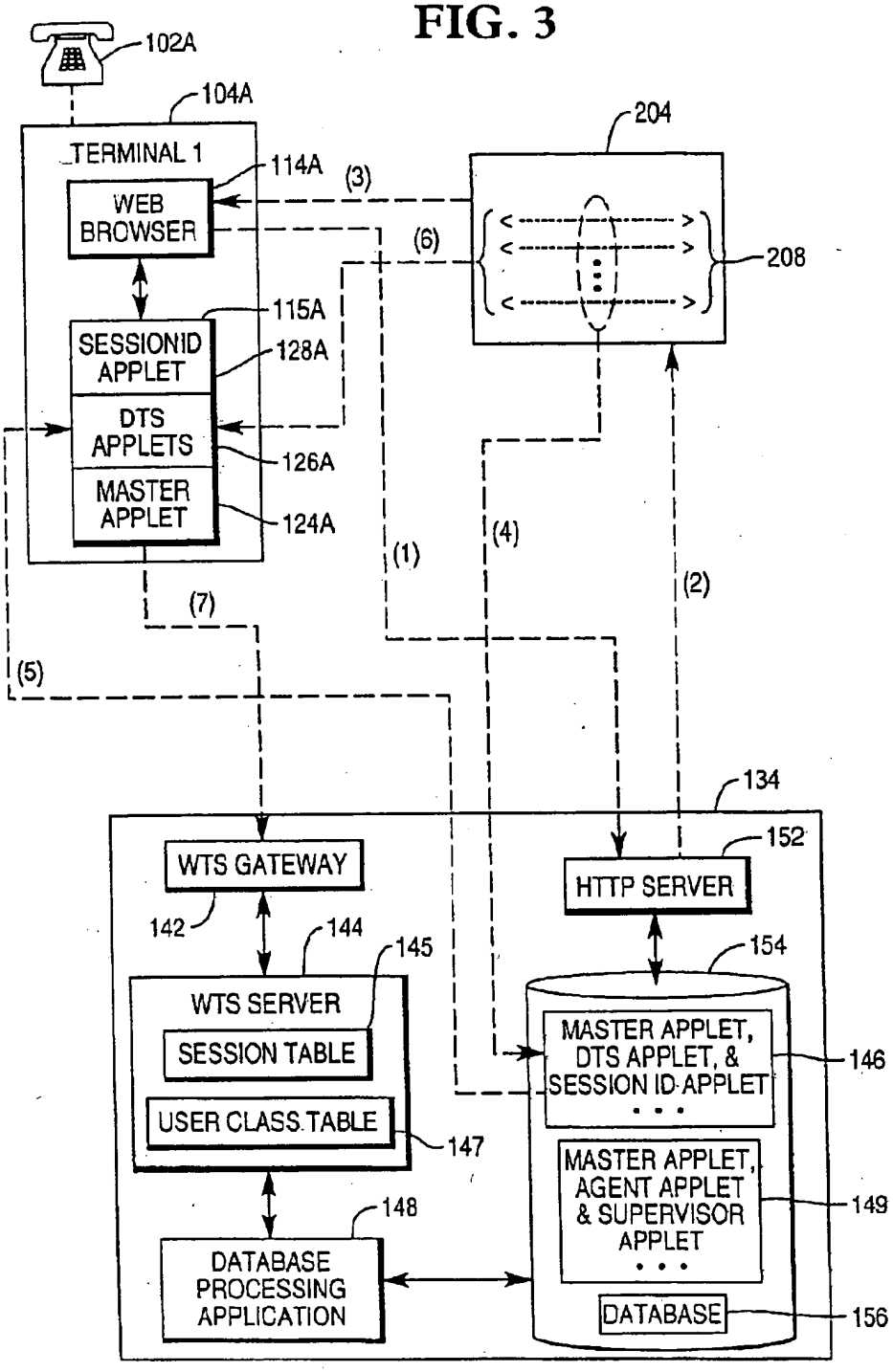

102A
104A
TERMINAL 1
114A
WEB BROWSER
(3)
204
(6)
208
115A
SESSIONID APPLET
128A
DTS APPLETS
126A
MASTER APPLET
124A
(1)
(4)
(2)
(7)
(5)
134
WTS GATEWAY
142
144
145
WTS SERVER
SESSION TABLE
USER CLASS TABLE
147
148
DATABASE PROCESSING APPLICATION
152
HTTP SERVER
154
MASTER APPLET, DTS APPLET, & SESSION ID APPLET
146
MASTER APPLET, AGENT APPLET & SUPERVISOR APPLET
149
DATABASE
156

FIG. 4

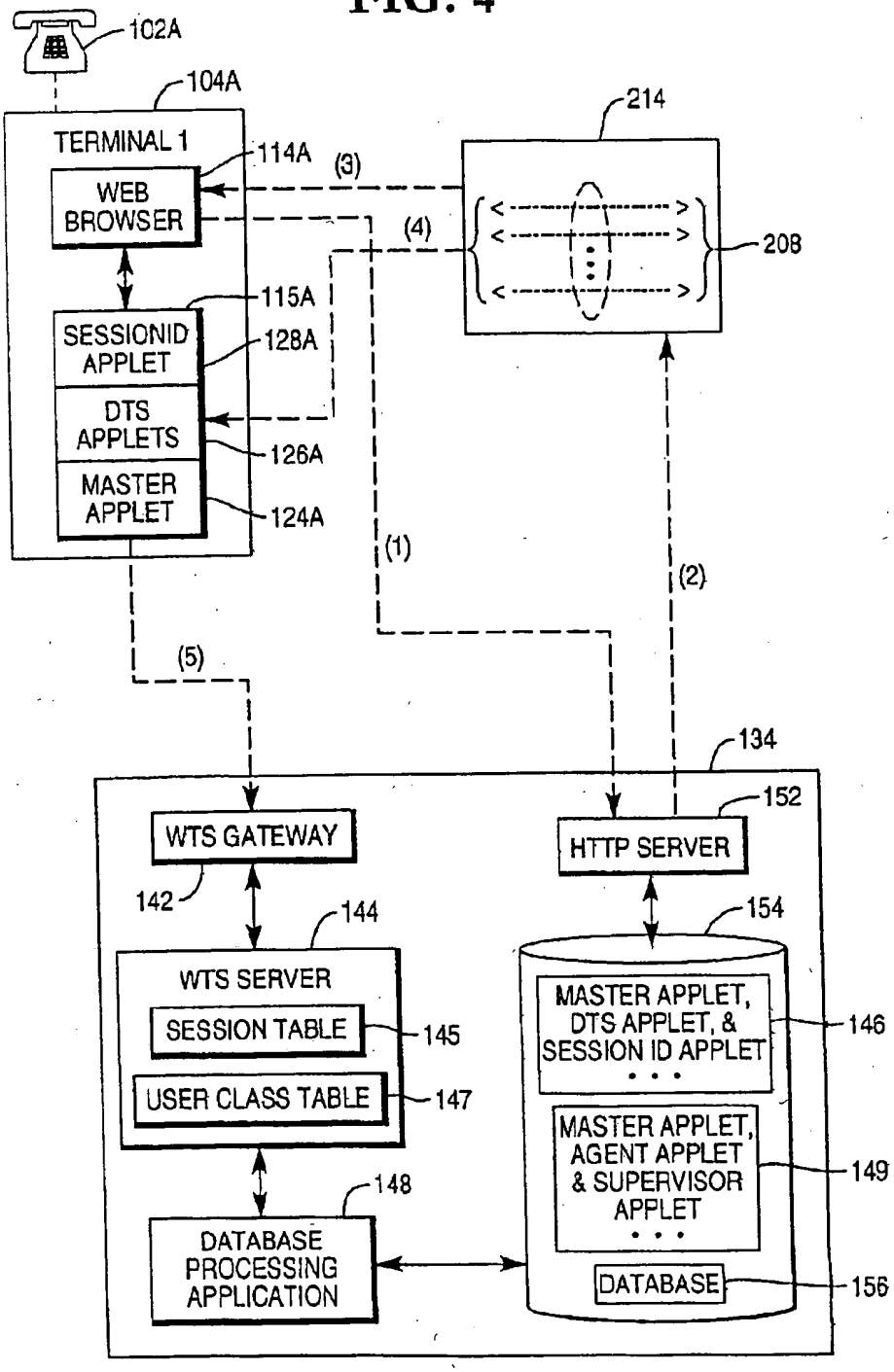

102A
104A
TERMINAL 1
114A
WEB BROWSER
(3)
(4)
214
208
115A
SESSIONID APPLET
128A
DTS APPLETS
126A
MASTER APPLET
124A
(1)
(2)
(5)
134
152
WTS GATEWAY
142
HTTP SERVER
144
154
WTS SERVER
SESSION TABLE
145
USER CLASS TABLE
147
MASTER APPLET, DTS APPLET, & SESSION ID APPLET
146
MASTER APPLET, AGENT APPLET & SUPERVISOR APPLET
149
148
DATABASE PROCESSING APPLICATION
DATABASE
156

FIG. 5

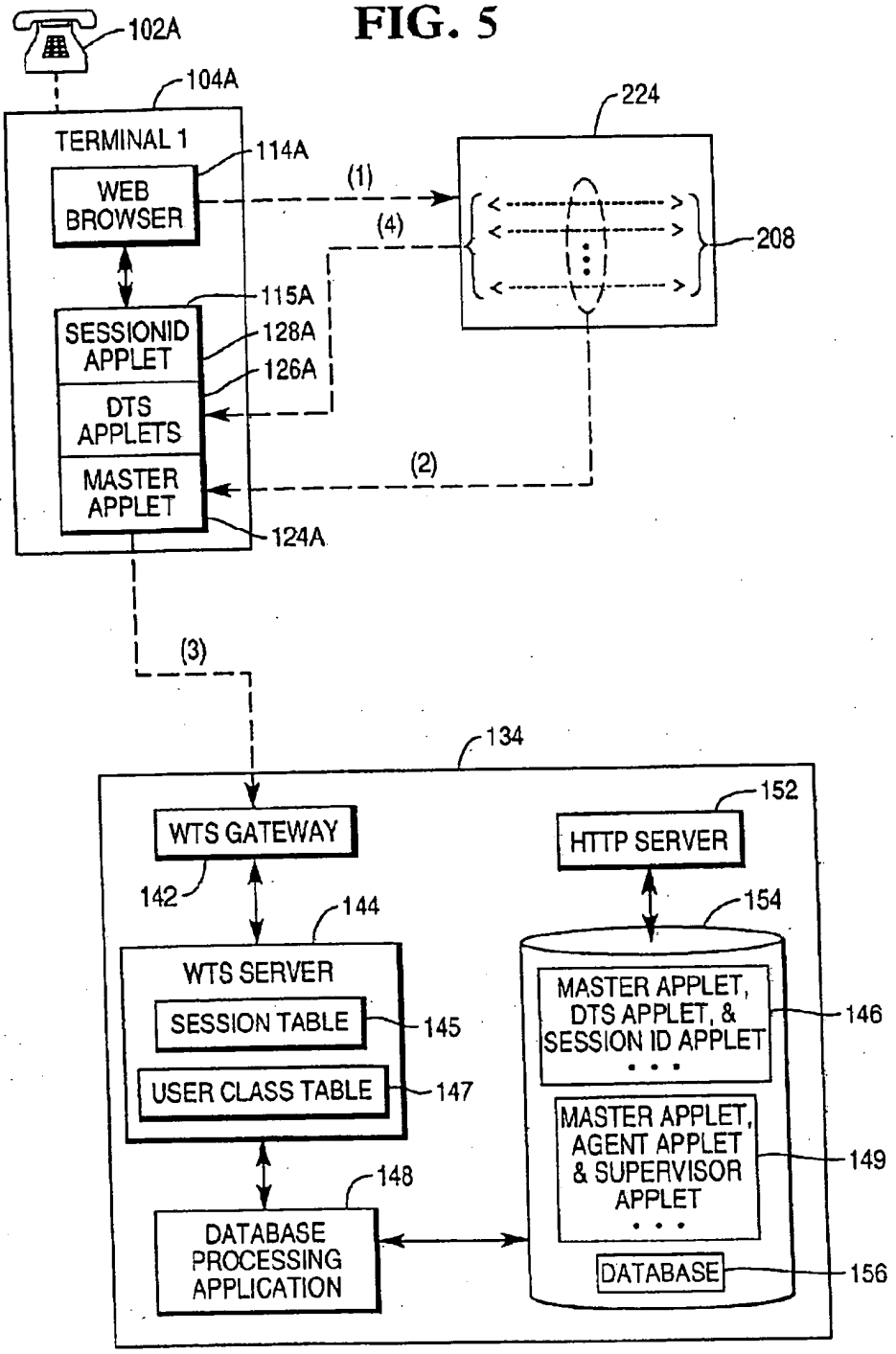

102A
104A
TERMINAL 1
114A
WEB BROWSER
(1)
224
208
(4)
115A
SESSIONID APPLET
128A
126A
DTS APPLETS
MASTER APPLET
(2)
124A
(3)
134
WTS GATEWAY
142
144
WTS SERVER
SESSION TABLE
145
USER CLASS TABLE
147
148
DATABASE PROCESSING APPLICATION
152
HTTP SERVER
154
MASTER APPLET, DTS APPLET, & SESSION ID APPLET
146
MASTER APPLET, AGENT APPLET & SUPERVISOR APPLET
149
DATABASE
156

FIG. 6

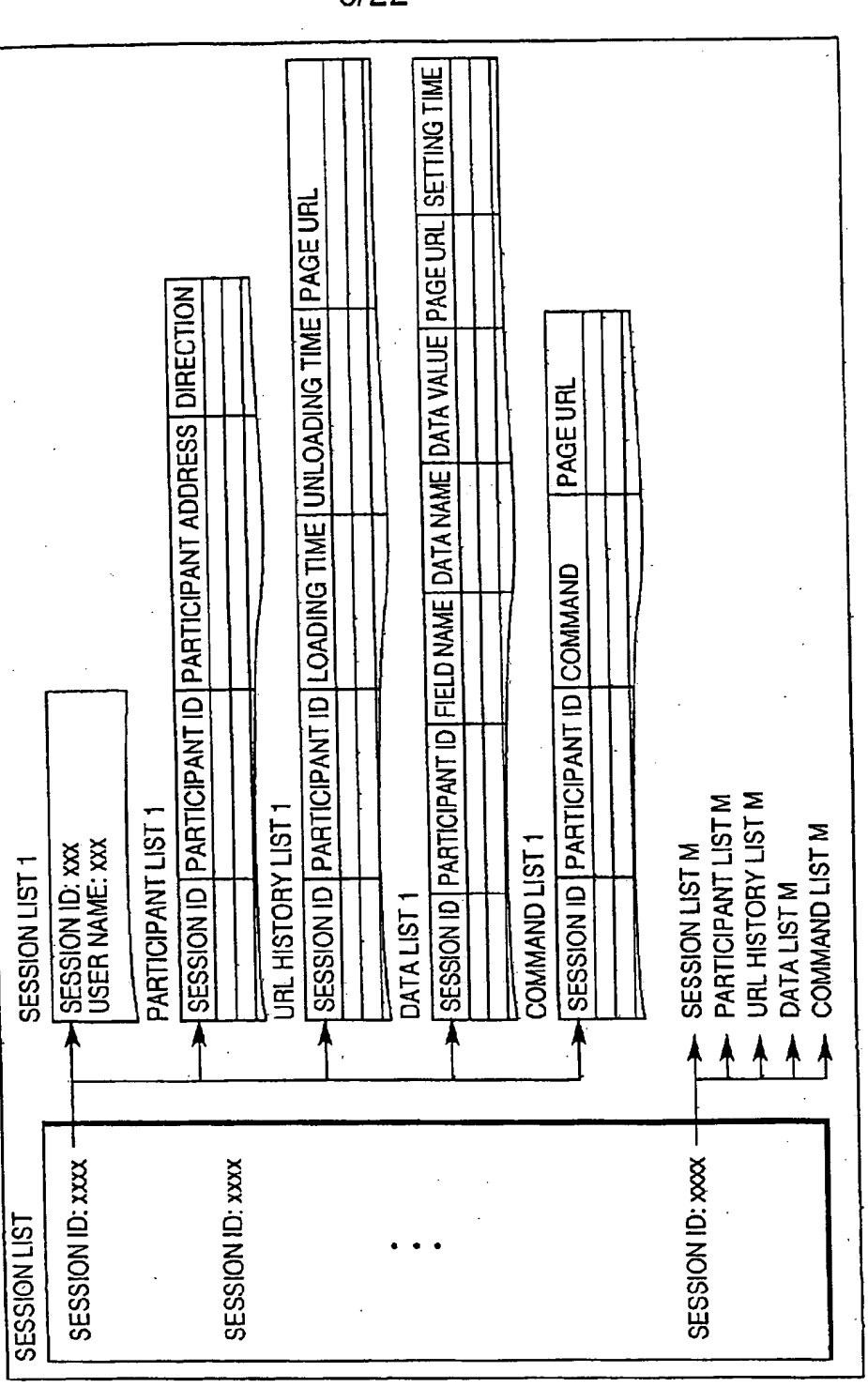

145
SESSION LIST
SESSION ID: xxxx
SESSION ID: xxxx
SESSION ID: xxxx
SESSION LIST 1
SESSION ID: xxx
USER NAME: xxx
PARTICIPANT LIST 1
SESSION ID
PARTICIPANT ID
PARTICIPANT ADDRESS
DIRECTION
URL HISTORY LIST 1
SESSION ID
PARTICIPANT ID
LOADING TIME
UNLOADING TIME
PAGE URL
DATA LIST 1
SESSION ID
PARTICIPANT ID
FIELD NAME
DATA NAME
DATA VALUE
PAGE URL
SETTING TIME
COMMAND LIST 1
SESSION ID
PARTICIPANT ID
COMMAND
PAGE URL
SESSION LIST M
PARTICIPANT LIST M
URL HISTORY LIST M
DATA LIST M
COMMAND LIST M

FIG. 7

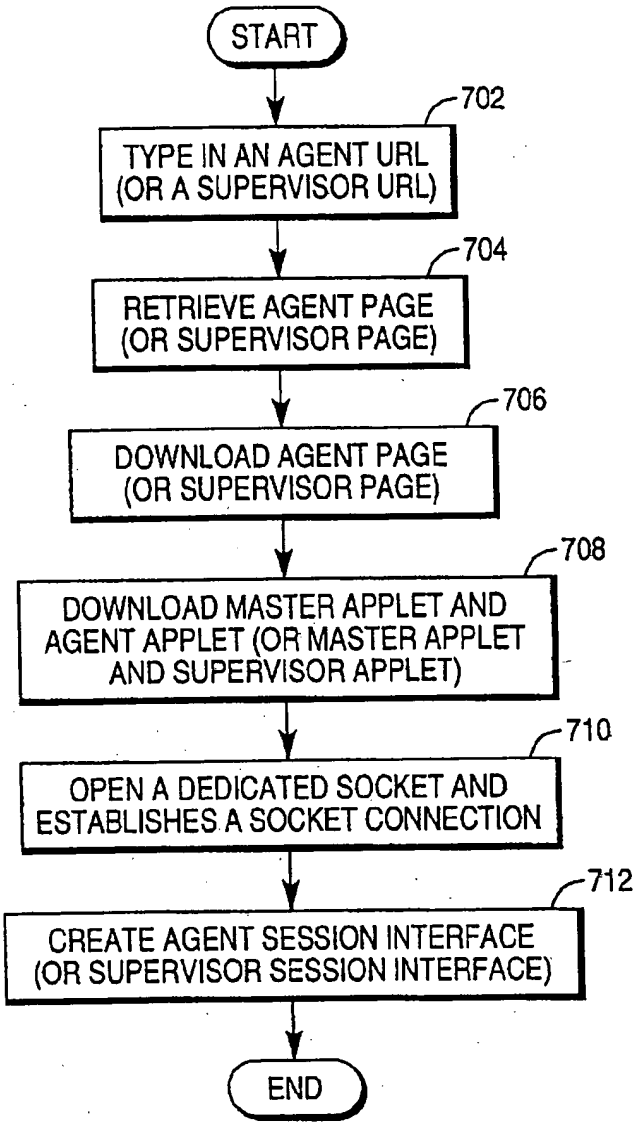

START
702
TYPE IN AN AGENT URL (OR A SUPERVISOR URL)
704
RETRIEVE AGENT PAGE (OR SUPERVISOR PAGE)
706
DOWNLOAD AGENT PAGE (OR SUPERVISOR PAGE)
708
DOWNLOAD MASTER APPLET AND AGENT APPLET (OR MASTER APPLET AND SUPERVISOR APPLET)
710
OPEN A DEDICATED SOCKET AND ESTABLISHES A SOCKET CONNECTION
712
CREATE AGENT SESSION INTERFACE (OR SUPERVISOR SESSION INTERFACE)
END

FIG. 8A

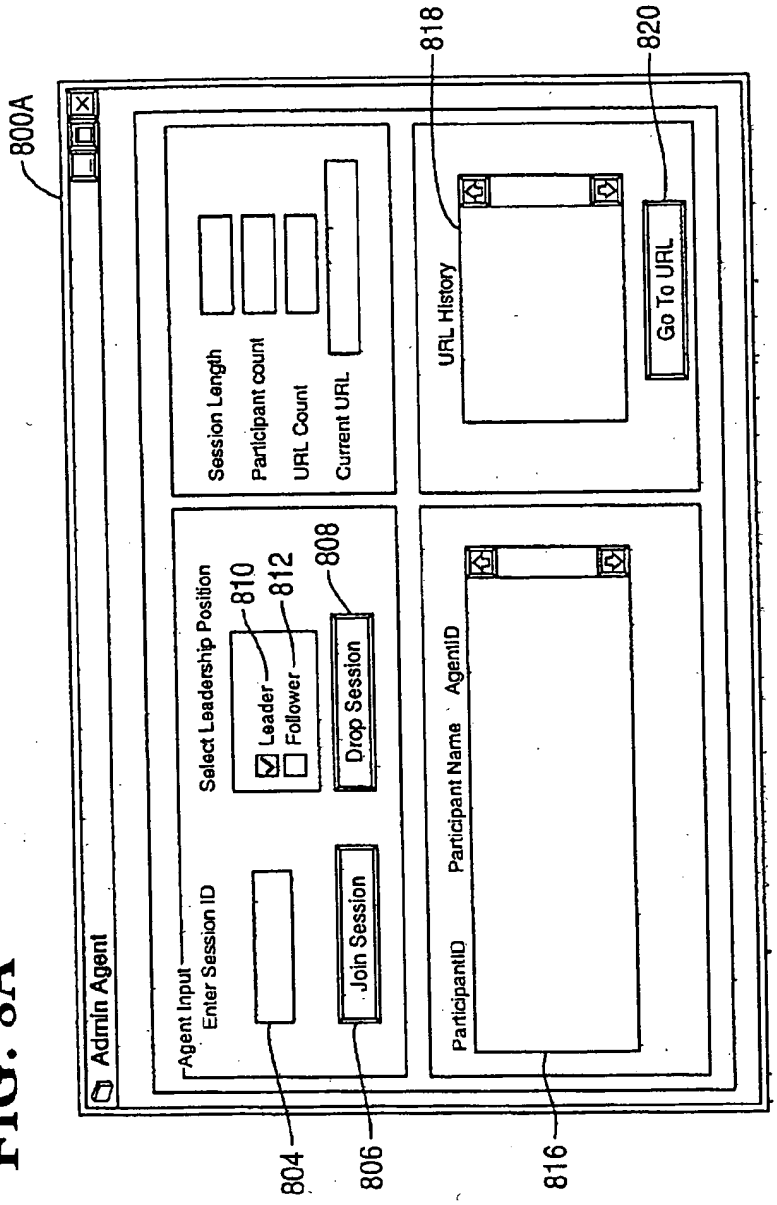

800A
Admin Agent
Agent Input
Enter Session ID
Select Leadership Position
Leader
Follower
Join Session
Drop Session
Session Length
Participant count
URL Count
Current URL
URL History
Go To URL
ParticipantID
Participant Name
AgentID
804
806
808
810
812
816
818
820

FIG. 8B

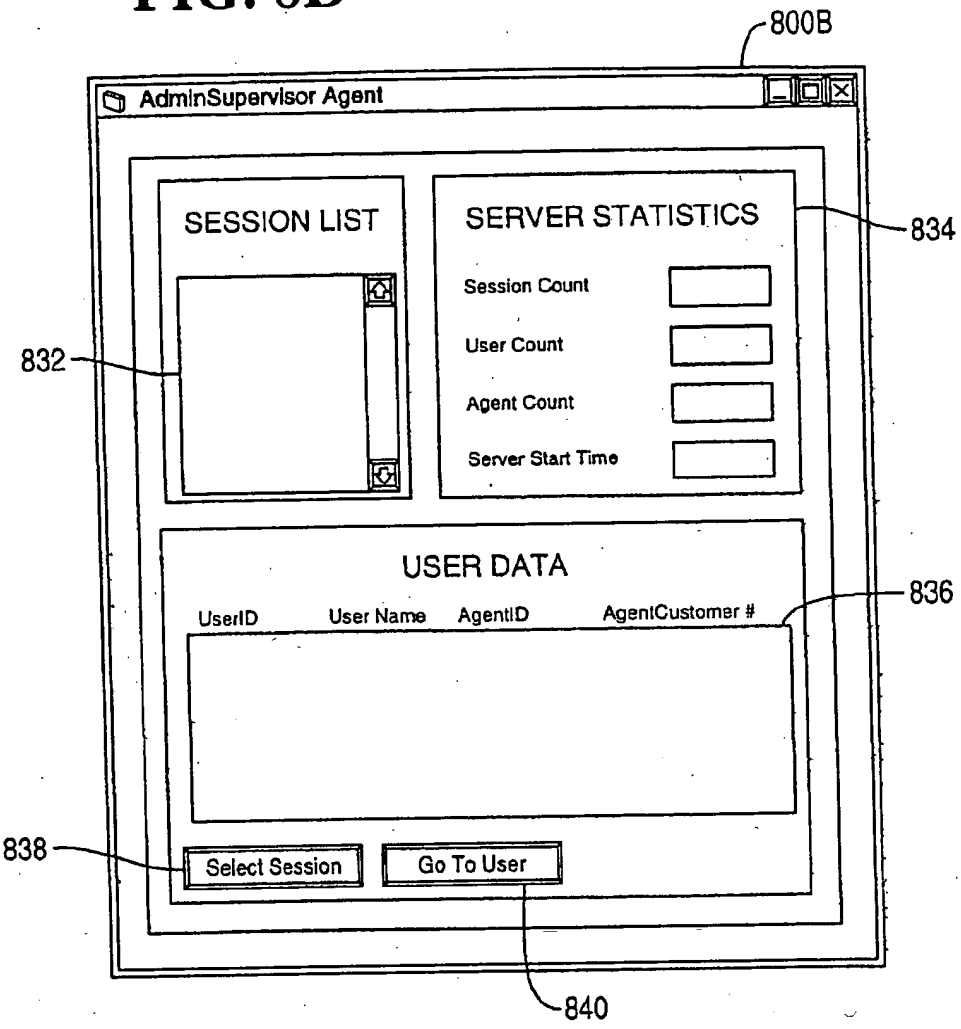

800B
AdminSupervisor Agent
SESSION LIST
SERVER STATISTICS
834
Session Count
User Count
Agent Count
Server Start Time
832
USER DATA
836
UserID
User Name
AgentID
AgentCustomer #
838
Select Session
Go To User
840

FIG. 8C

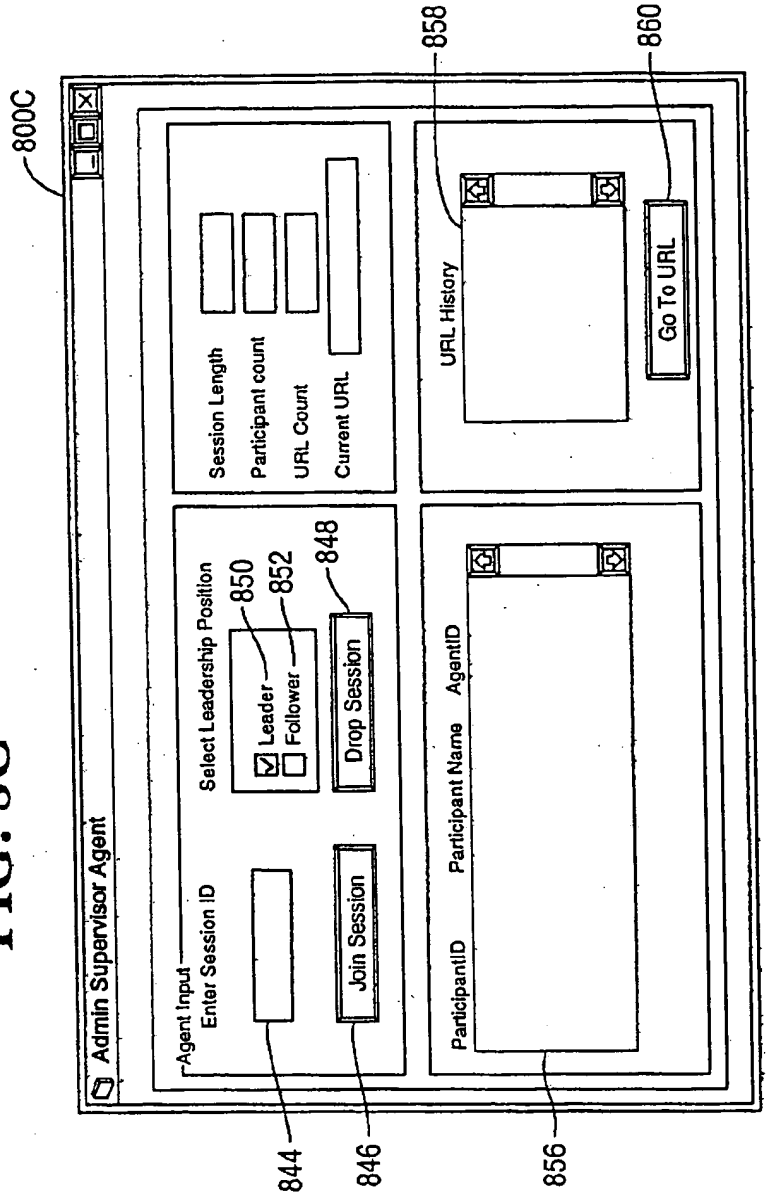

800C
858
860
Admin Supervisor Agent
Session Length
Participant count
URL Count
Current URL
URL History
Go To URL
Agent Input
Enter Session ID
Select Leadership Position
850
852
848
Leader
Follower
Drop Session
Join Session
ParticipantID
Participant Name
AgentID
844
846
856

FIG. 9

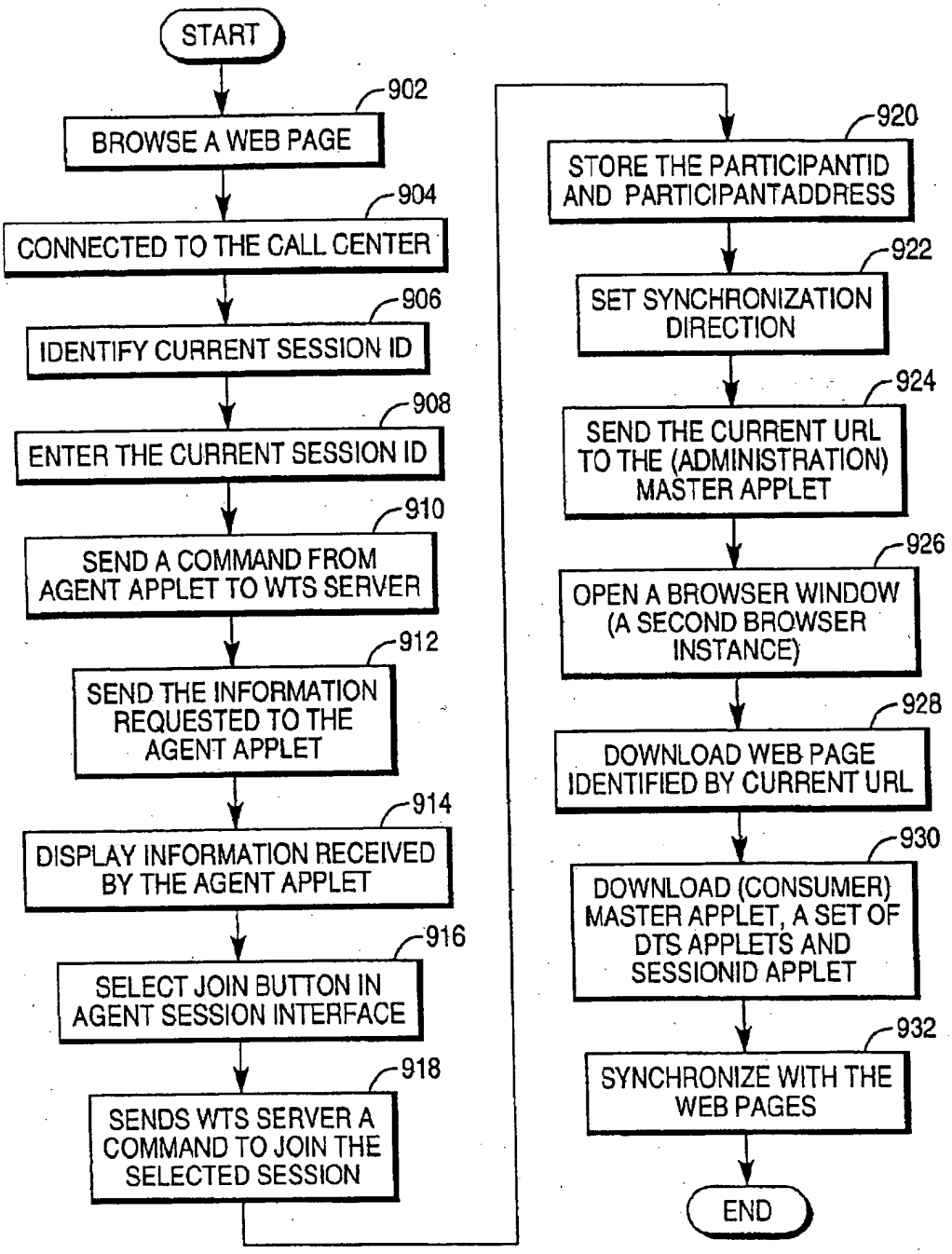

START
902
BROWSE A WEB PAGE
904
CONNECTED TO THE CALL CENTER
906
IDENTIFY CURRENT SESSION ID
908
ENTER THE CURRENT SESSION ID
910
SEND A COMMAND FROM AGENT APPLET TO WTS SERVER
912
SEND THE INFORMATION REQUESTED TO THE AGENT APPLET
914
DISPLAY INFORMATION RECEIVED BY THE AGENT APPLET
916
SELECT JOIN BUTTON IN AGENT SESSION INTERFACE
918
SENDS WTS SERVER A COMMAND TO JOIN THE SELECTED SESSION
920
STORE THE PARTICIPANTID AND PARTICIPANTADDRESS
922
SET SYNCHRONIZATION DIRECTION
924
SEND THE CURRENT URL TO THE (ADMINISTRATION) MASTER APPLET
926
OPEN A BROWSER WINDOW (A SECOND BROWSER INSTANCE)
928
DOWNLOAD WEB PAGE IDENTIFIED BY CURRENT URL
930
DOWNLOAD (CONSUMER) MASTER APPLET, A SET OF DTS APPLETS AND SESSIONID APPLET
932
SYNCHRONIZE WITH THE WEB PAGES
END

FIG. 10

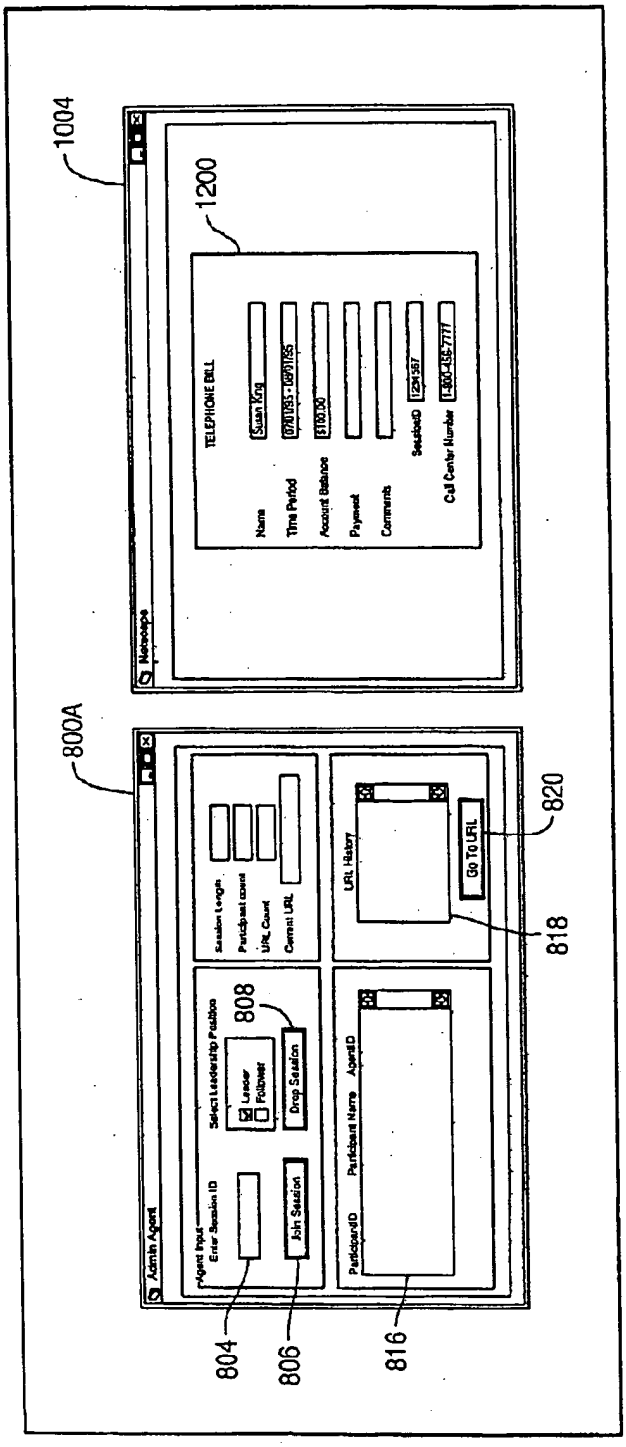

1004
1200
TELEPHONE BILL
Name
Time Period
Account Balance
Payment
Comments
Call Center Number
800A
Admin Agent
Go To URL
URL History
808
804
806
816
818
820

FIG. 11

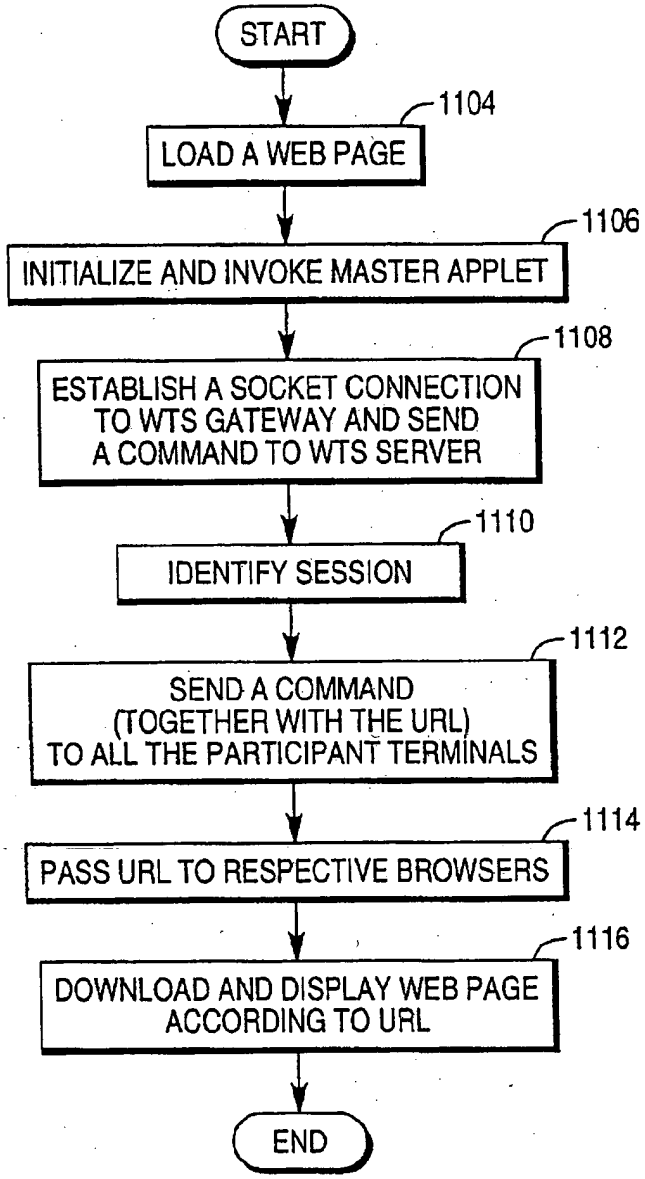

START
1104
LOAD A WEB PAGE
1106
INITIALIZE AND INVOKE MASTER APPLET
1108
ESTABLISH A SOCKET CONNECTION TO WTS GATEWAY AND SEND A COMMAND TO WTS SERVER
1110
IDENTIFY SESSION
1112
SEND A COMMAND (TOGETHER WITH THE URL) TO ALL THE PARTICIPANT TERMINALS
1114
PASS URL TO RESPECTIVE BROWSERS
1116
DOWNLOAD AND DISPLAY WEB PAGE ACCORDING TO URL
END

FIG. 12A

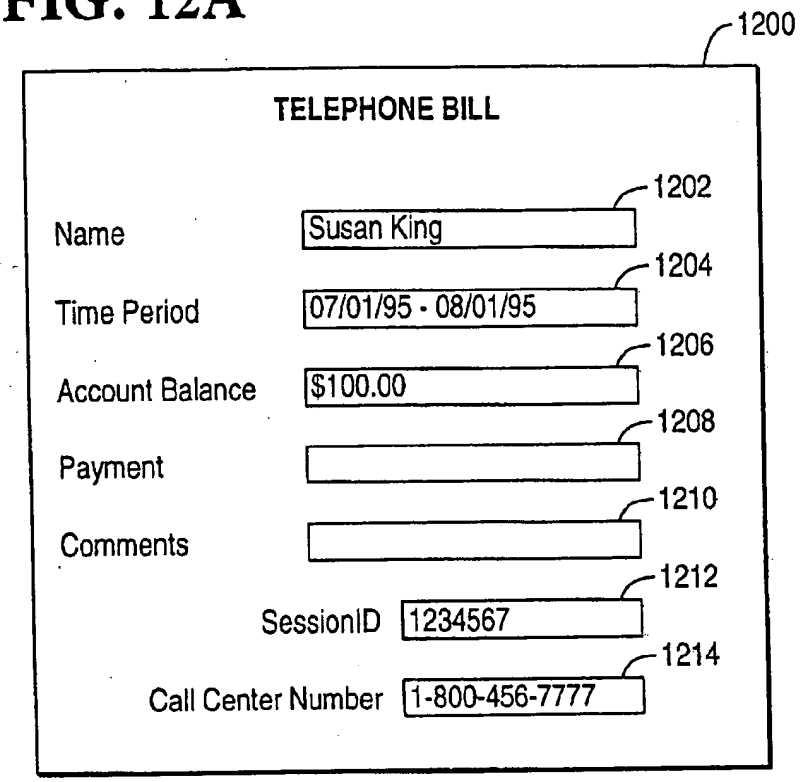

1200
TELEPHONE BILL
1202
Name
Susan King
1204
Time Period
07/01/95 - 08/01/95
1206
Account Balance
$100.00
1208
Payment
1210
Comments
1212
SessionID
1234567
1214
Call Center Number
1-800-456-7777

FIG. 12B

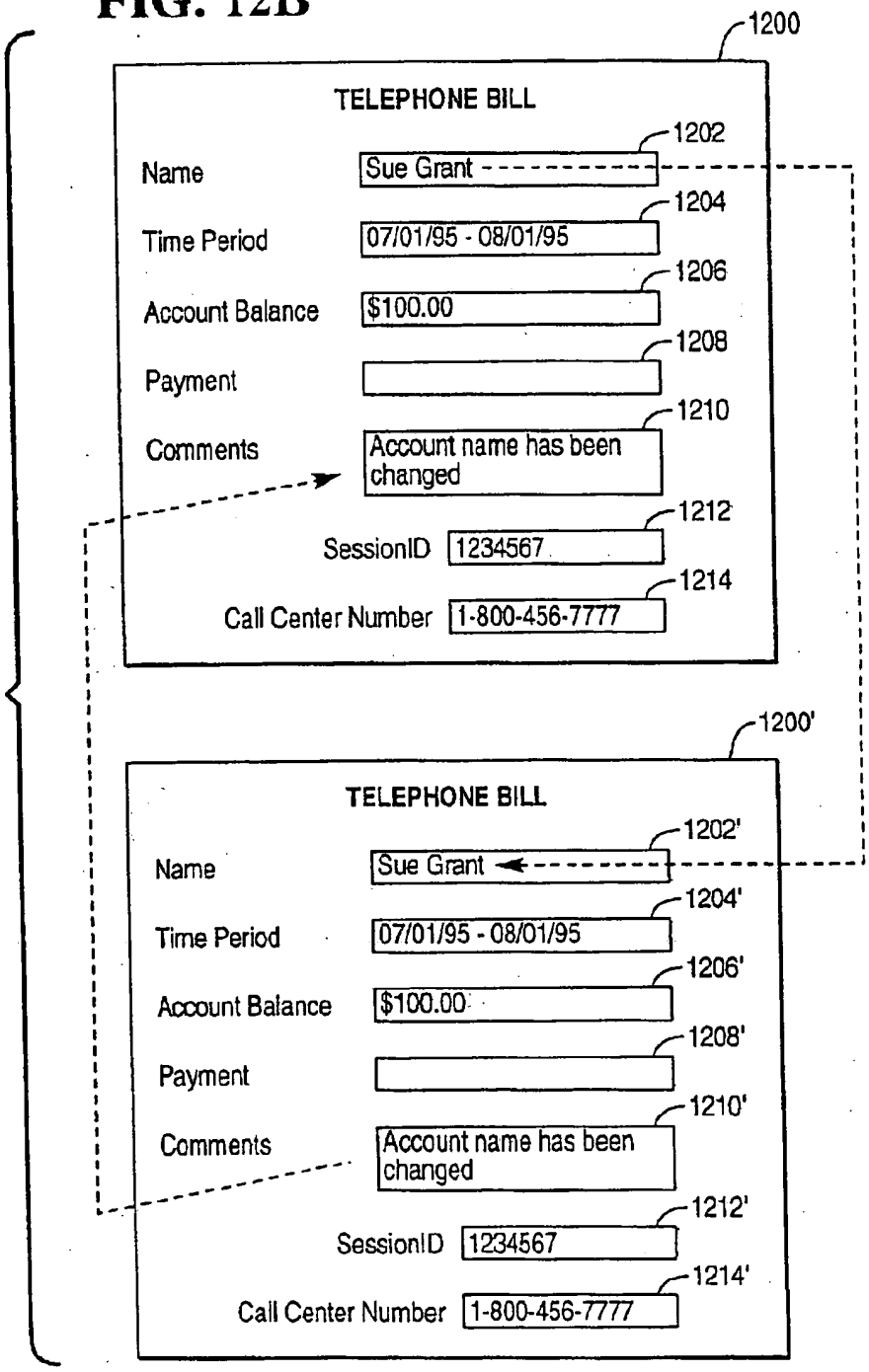

1200
TELEPHONE BILL
1202
Name
Sue Grant
1204
Time Period
07/01/95 - 08/01/95
1206
Account Balance
$100.00
1208
Payment
1210
Comments
Account name has been changed
1212
SessionID
1234567
1214
Call Center Number
1-800-456-7777
1200'
TELEPHONE BILL
1202'
Name
Sue Grant
1204'
Time Period
07/01/95 - 08/01/95
1206'
Account Balance
$100.00
1208'
Payment
1210'
Comments
Account name has been changed
1212'
SessionID
1234567
1214'
Call Center Number
1-800-456-7777

FIG. 13

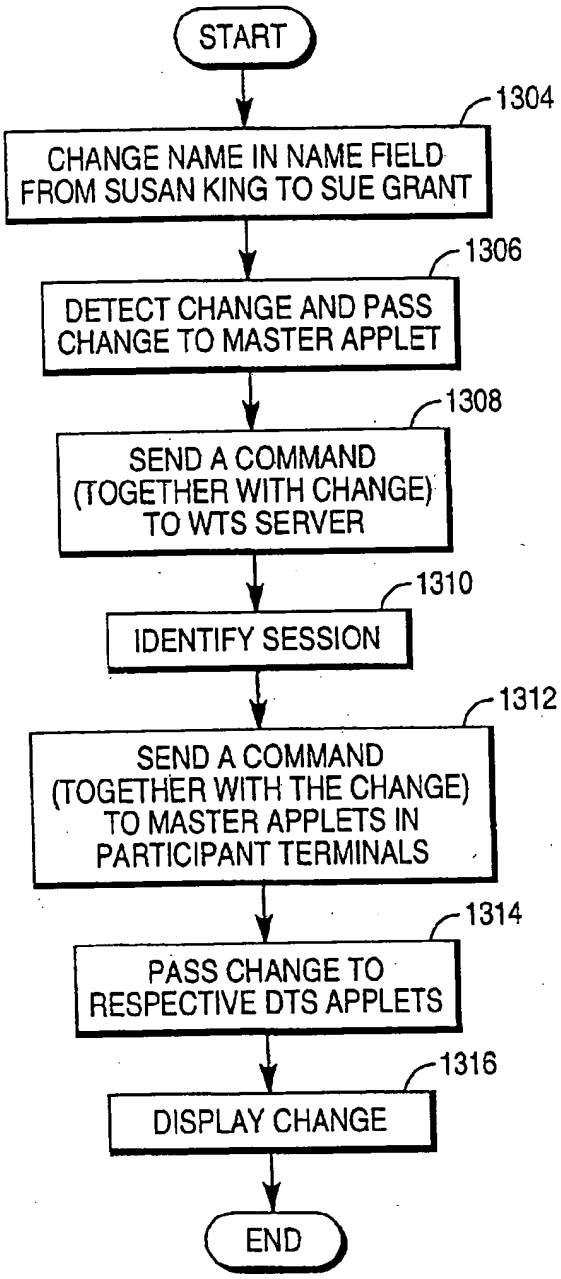

START
1304
CHANGE NAME IN NAME FIELD FROM SUSAN KING TO SUE GRANT
1306
DETECT CHANGE AND PASS CHANGE TO MASTER APPLET
1308
SEND A COMMAND (TOGETHER WITH CHANGE) TO WTS SERVER
1310
IDENTIFY SESSION
1312
SEND A COMMAND (TOGETHER WITH THE CHANGE) TO MASTER APPLETS IN PARTICIPANT TERMINALS
1314
PASS CHANGE TO RESPECTIVE DTS APPLETS
1316
DISPLAY CHANGE
END

FIG. 14

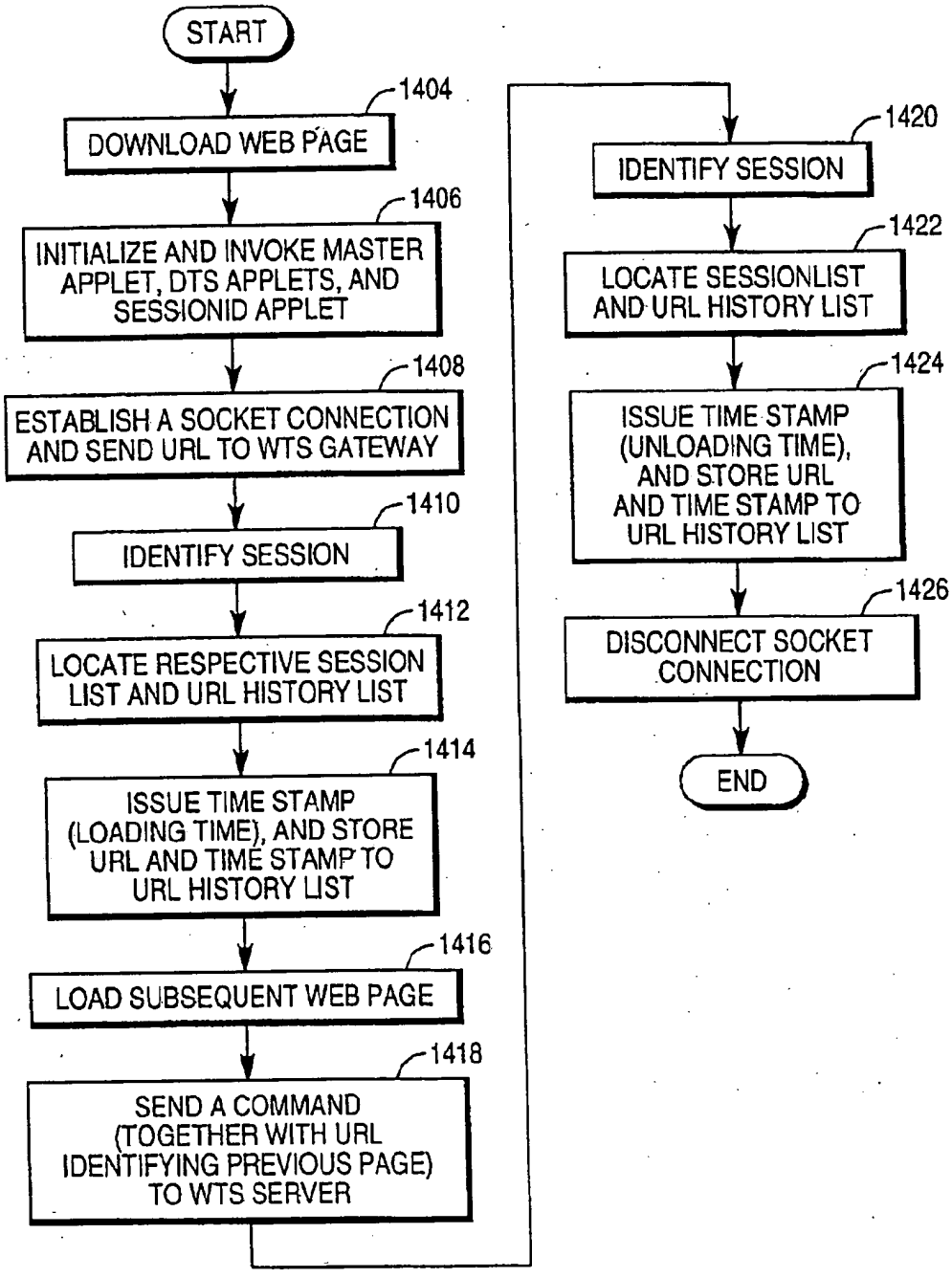

START
1404
DOWNLOAD WEB PAGE
1406
INITIALIZE AND INVOKE MASTER APPLET, DTS APPLETS, AND SESSIONID APPLET
1408
ESTABLISH A SOCKET CONNECTION AND SEND URL TO WTS GATEWAY
1410
IDENTIFY SESSION
1412
LOCATE RESPECTIVE SESSION LIST AND URL HISTORY LIST
1414
ISSUE TIME STAMP (LOADING TIME), AND STORE URL AND TIME STAMP TO URL HISTORY LIST
1416
LOAD SUBSEQUENT WEB PAGE
1418
SEND A COMMAND (TOGETHER WITH URL IDENTIFYING PREVIOUS PAGE) TO WTS SERVER
1420
IDENTIFY SESSION
1422
LOCATE SESSIONLIST AND URL HISTORY LIST
1424
ISSUE TIME STAMP (UNLOADING TIME), AND STORE URL AND TIME STAMP TO URL HISTORY LIST
1426
DISCONNECT SOCKET CONNECTION
END

FIG. 15

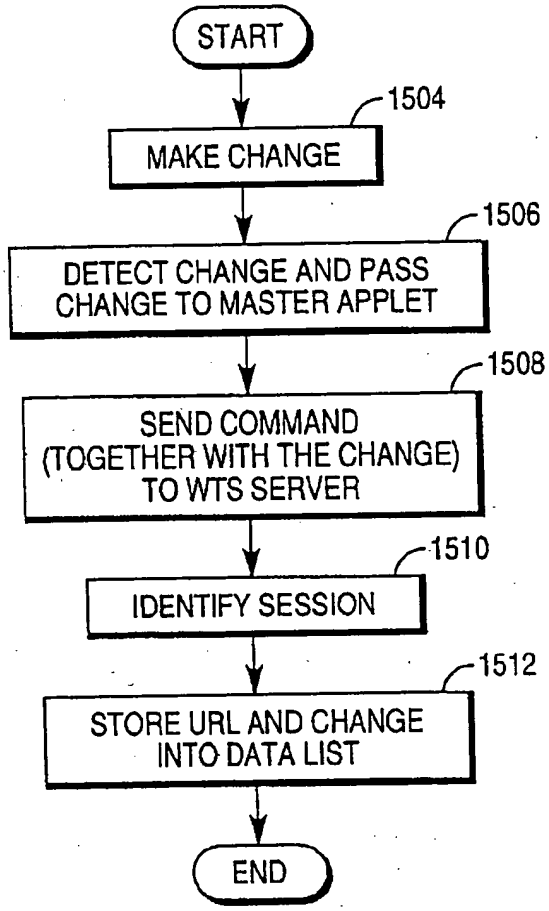

START
1504
MAKE CHANGE
1506
DETECT CHANGE AND PASS CHANGE TO MASTER APPLET
1508
SEND COMMAND (TOGETHER WITH THE CHANGE) TO WTS SERVER
1510
IDENTIFY SESSION
1512
STORE URL AND CHANGE INTO DATA LIST
END

FIG. 16

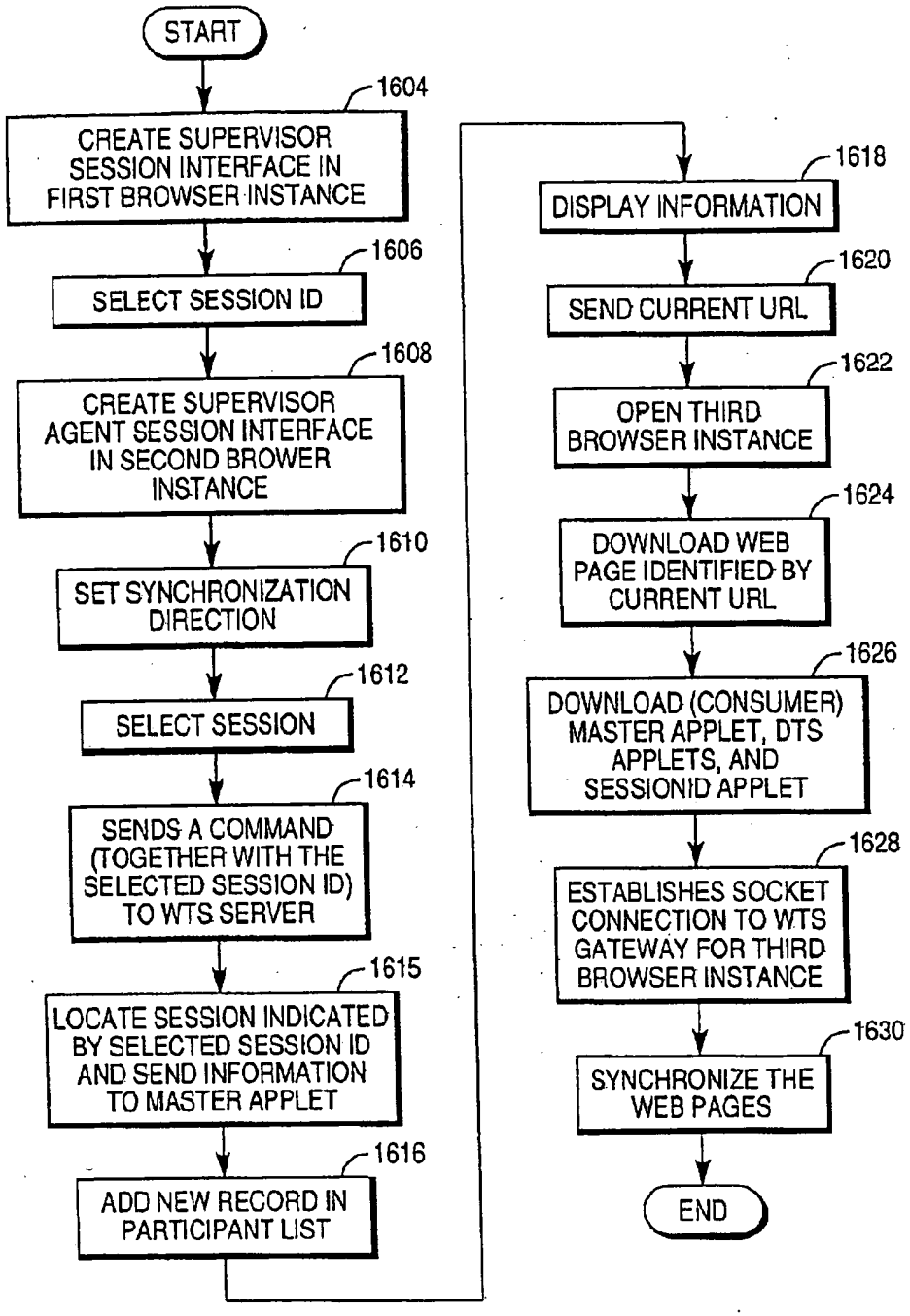

START
1604
CREATE SUPERVISOR SESSION INTERFACE IN FIRST BROWSER INSTANCE
1606
SELECT SESSION ID
1608
CREATE SUPERVISOR AGENT SESSION INTERFACE IN SECOND BROWER INSTANCE
1610
SET SYNCHRONIZATION DIRECTION
1612
SELECT SESSION
1614
SENDS A COMMAND (TOGETHER WITH THE SELECTED SESSION ID) TO WTS SERVER
1615
LOCATE SESSION INDICATED BY SELECTED SESSION ID AND SEND INFORMATION TO MASTER APPLET
1616
ADD NEW RECORD IN PARTICIPANT LIST
1618
DISPLAY INFORMATION
1620
SEND CURRENT URL
1622
OPEN THIRD BROWSER INSTANCE
1624
DOWNLOAD WEB PAGE IDENTIFIED BY CURRENT URL
1626
DOWNLOAD (CONSUMER) MASTER APPLET, DTS APPLETS, AND SESSIONID APPLET
1628
ESTABLISHES SOCKET CONNECTION TO WTS GATEWAY FOR THIRD BROWSER INSTANCE
1630
SYNCHRONIZE THE WEB PAGES
END

FIG. 17

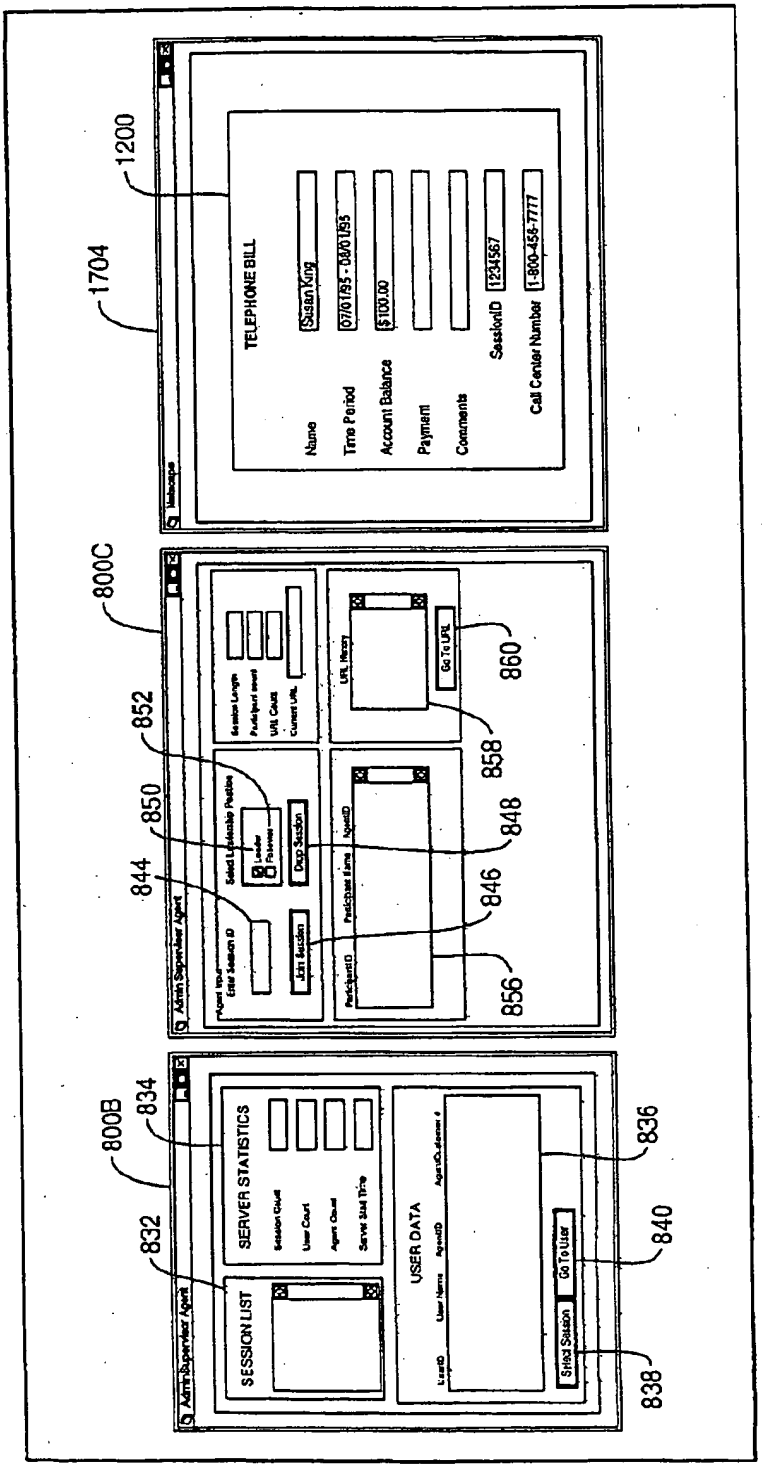

1200
1704
800C
800B
834
832
852
850
844
848
846
856
858
860
836
840
838
TELEPHONE BILL
Name
Susan King
Time Period
07/01/95 - 08/01/95
Account Balance
$100.00
Payment
Comments
SessionID
1234567
Call Center Number
1-800-458-7777
SERVER STATISTICS
SESSION LIST
USER DATA
Join Session
Drop Session
Go To URL
Send Session
Go To User

FIG. 18

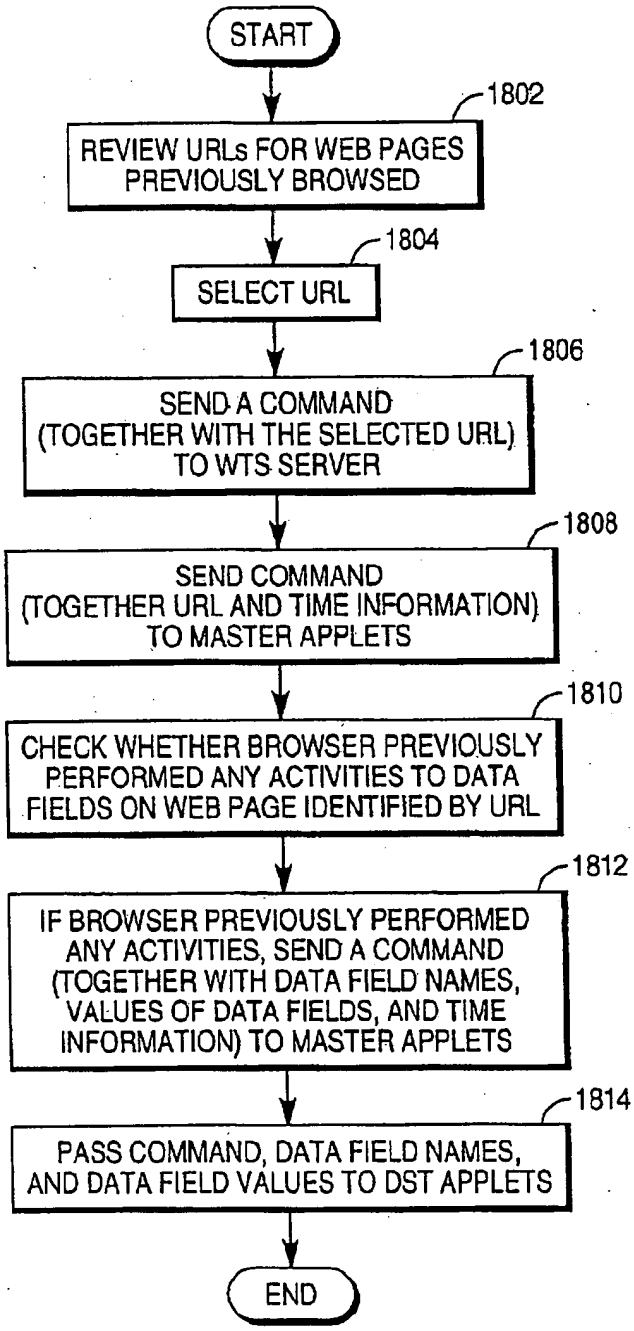
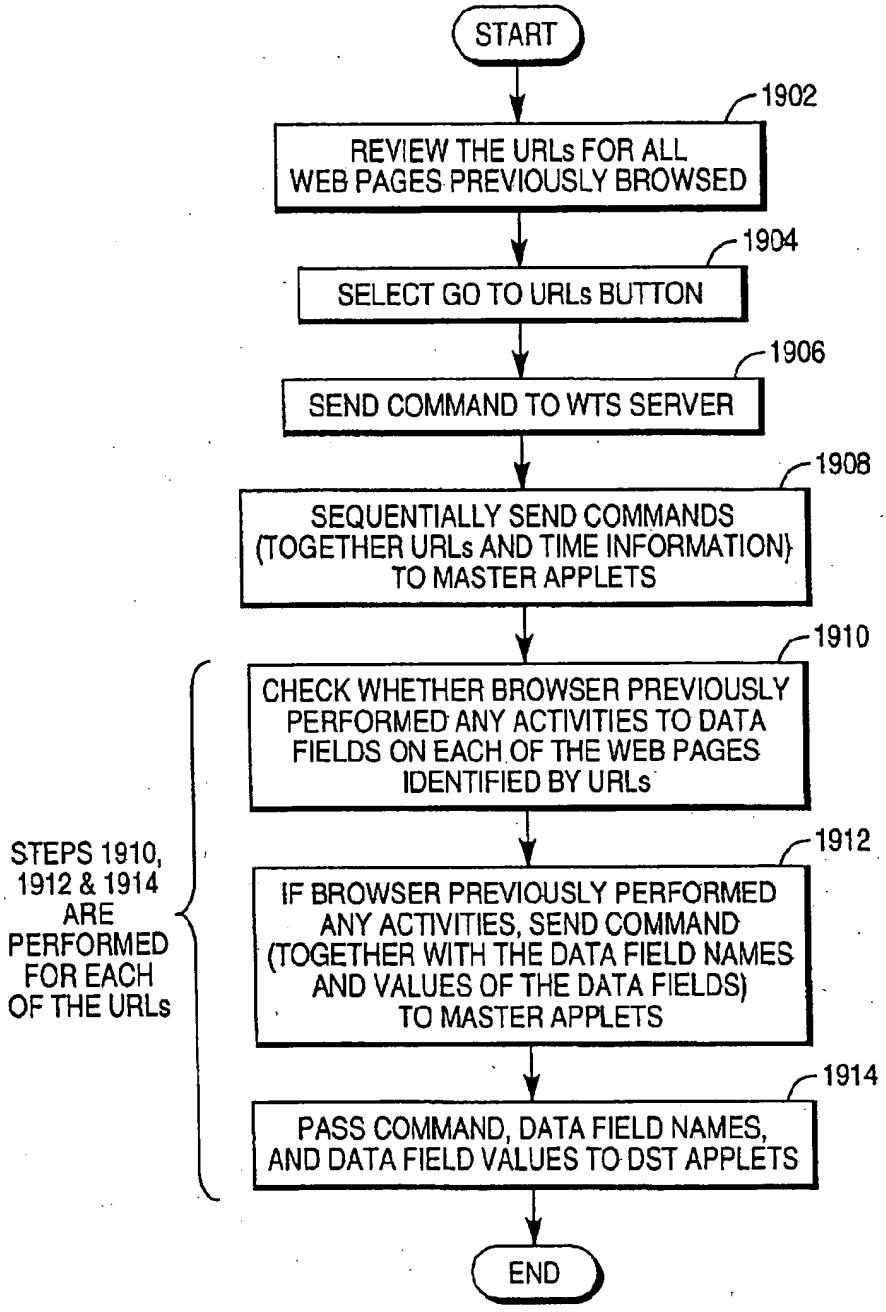

START
1802
REVIEW URLs FOR WEB PAGES PREVIOUSLY BROWSED
1804
SELECT URL
1806
SEND A COMMAND (TOGETHER WITH THE SELECTED URL) TO WTS SERVER
1808
SEND COMMAND (TOGETHER URL AND TIME INFORMATION) TO MASTER APPLETS
1810
CHECK WHETHER BROWSER PREVIOUSLY PERFORMED ANY ACTIVITIES TO DATA FIELDS ON WEB PAGE IDENTIFIED BY URL
1812
IF BROWSER PREVIOUSLY PERFORMED ANY ACTIVITIES, SEND A COMMAND (TOGETHER WITH DATA FIELD NAMES, VALUES OF DATA FIELDS, AND TIME INFORMATION) TO MASTER APPLETS
1814
PASS COMMAND, DATA FIELD NAMES, AND DATA FIELD VALUES TO DST APPLETS
END

FIG. 19


START
1902
REVIEW THE URLs FOR ALL WEB PAGES PREVIOUSLY BROWSED
1904
SELECT GO TO URLs BUTTON
1906
SEND COMMAND TO WTS SERVER
1908
SEQUENTIALLY SEND COMMANDS (TOGETHER URLs AND TIME INFORMATION) TO MASTER APPLETS
1910
CHECK WHETHER BROWSER PREVIOUSLY PERFORMED ANY ACTIVITIES TO DATA FIELDS ON EACH OF THE WEB PAGES IDENTIFIED BY URLs
1912
IF BROWSER PREVIOUSLY PERFORMED ANY ACTIVITIES, SEND COMMAND (TOGETHER WITH THE DATA FIELD NAMES AND VALUES OF THE DATA FIELDS) TO MASTER APPLETS
1914
PASS COMMAND, DATA FIELD NAMES, AND DATA FIELD VALUES TO DST APPLETS
END
STEPS 1910, 1912 & 1914 ARE PERFORMED FOR EACH OF THE URLs

1 Abstract

A mechanism for dependably tracking network activity such as web page activity among terminals [104A..104K...104N] such as a group of browsers. The terminals retrieve pages [204...214...224] from an HTTP server [152], each of the pages having embedded information [208] such as an applet. In response to the page activity (such as loading, unloading or changing data [Fig12A, Fig 12B]) performed at a terminal the respective applet [124, 126, 128] reports the activity (together with the page URL) to a server [144] which in turn stores it in a database [148]. Synchronization between pages at different terminals is provided by appropriate embedded information, such as a data tracking and synchronizing applet [126], which identifies activities and forwards details of the activities via the terminal to a tracking server for database storage and transmission to other participating terminals as required. A session [Fig.6] can be created for an individual terminal or a group of terminals and terminal activity information held as part of the session information for access as required by other terminals in or joining the session. one of the terminals may be an administrative terminal.

2 Representative Drawing

FIG. 1